AD-A040 945   INSTITUTE FOR DEFENSE ANALYSES ARLINGTON VA PROGRAM --ETC F/G 15/4
DOCUMENTATION OF CURRENT IDA COMPUTER MATERIAL DEVELOPED FOR DC--ETC(U)
JAN 77 L A SCHMIDT   DCPA01-76-C-0213

UNCLASSIFIED   IDA/HQ-77-19225   NL

1 OF 3

AD
A040945

4

B.S.

DOCUMENTATION OF CURRENT IDA COMPUTER MATERIAL DEVELOPED FOR DCPA

Volume I

Leo A. Schmidt

January 1977

COPY AVAILABLE TO DDC DOES NOT
PERMIT FULLY LEGIBLE PRODUCTION

INSTITUTE FOR DEFENSE ANALYSES
Program Analysis Division
Contract No. DCPA01-76-C-0213
Work Unit 4126G

IDA Log No. HQ 77-19225
Copy 3 of 14 copies

The work reported in this document was conducted under Contract No. DCPA01-76-C-0213 for the Defense Civil Preparedness Agency. The publication of this document does not indicate endorsement by the Defense Civil Preparedness Agency, nor should the contents be construed as reflecting the official position of that agency.

Approved for public release; distribution unlimited.

DOCUMENTATION OF CURRENT IDA COMPUTER MATERIAL DEVELOPED FOR DCPA

Volume I

by

Leo A. Schmidt

for

Defense Civil Preparedness Agency
Washington, D.C. 20301

DCPA REVIEW NOTICE

This report has been reviewed in the Defense Civil Preparedness Agency and approved for publication. Approval does not signify that the contents necessarily reflect the views and policies of the Defense Civil Preparedness Agency.

INSTITUTE FOR DEFENSE ANALYSES
Program Analysis Division
400 Army-Navy Drive, Arlington, Virginia 22202

Contract No. DCPA01-76-C-0213
Work Unit 4126G

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NA	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 Documentation of Current IDA Computer Material Developed for DCPA, Volume I,		5. TYPE OF REPORT & PERIOD COVERED NA
		6. PERFORMING ORG. REPORT NUMBER NA
7. AUTHOR(s) 10 Leo A. Schmidt	15	8. CONTRACT OR GRANT NUMBER(s) DCPA01-76-C-0213
9. PERFORMING ORGANIZATION NAME AND ADDRESS INSTITUTE FOR DEFENSE ANALYSES Program Analysis Division 400 Army-Navy Drive, Arlington, VA		10. PROGRAM ELEMENT PROJECT TASK AREA & WORK UNIT NUMBERS Work Unit ~~4126G~~ 4114G
11. CONTROLLING OFFICE NAME AND ADDRESS DEFENSE CIVIL PREPAREDNESS AGENCY Washington, D.C. 20301	11	12. REPORT DATE Jan~~uary~~ 1977
		13. NUMBER OF PAGES Vol. I--283 pages
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12 184 p.		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18 IDA/HQ 77-19225

18. SUPPLEMENTARY NOTES

19

Vol 2 - A037 819

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Damage Assessment

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper is a documentation of computer materials developed by the Institute for Defense Analyses (IDA) for use by the Defense Civil Preparedness Agency (DCPA). All IDA physical data processing materials (IBM cards, magnetic tape, computer printouts) have been surveyed and catalogued. All computer programs are written in FORTRAN (a general knowledge of this language is assumed in the detailed descriptions contained herein). Computer programs considered useful by IDA have been included and

→ next page

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

403 219 ✓

20. continued

documented. A group of general purpose subprograms are described, along with their interfaces with the using programs. Data file formats also have been developed, along with programs for managing these files. Such programs and resulting files are described in detail.

CONTENTS

ABSTRACT vi

Volume I

I. INTRODUCTION I-1

II. COMPUTER PROGRAMS I-5

DISCRN I-6

SWOOSH I-7

DWNFIT I-8

WINDST I-9

LASH I-10

LASHCO I-31

COLPLT I-32

RUBATO I-33

AGZSEL I-83

LANCET I-84

AIP/ANCET I-85

ALLEGRO I-86

ANDANTE I-87

GEM/PADECON I-135

TOGEM I-136

GEMLET I-137

MARATHON I-138

RESVAL I-139

ADAGIO I-151

TSTGD I-152

JLAM I-153

TEST I-154

POPPOP I-155

FIRESM I-163

MESHFL I-173

A. Introduction I-174

B. Subroutine MEET I-176

C. Subroutine MERGE I-176

D. Subroutine EDIT I-179

SNORT I-229
BOX I-235
MOVEUB I-247
SHUFFL I-251
CRDATK I-255
TOGRID I-259
LLIST I-273
WPGRID I-277

Volume II

SUBPROGRAMS II-1
Subroutine CTYDAM II-3
Subroutine NEWCTY II-19
Subroutine OPTWPN II-25
Subroutine ONEPAS II-35
Subroutine FLPKA II-39
Subroutine FLPKHU II-49
Function PZZ II-57
Function SSKP II-61
Function G II-65
Function XLRAD II-69
Function ROOTF II-73
Subroutine CALRN II-79
Function CUMNOR II-89
Subroutine FORD II-93
Subroutine PROJCT II-97
Subroutine BIMOM II-130
Subroutine TELFAR II-153
Subroutines FALLYB, FALLWB, FALLDB, FALLCB . . II-161
Subroutines FALLYQ, FALLWQ, FALLDQ, FALLCQ . . II-181
Subroutine CFALLY II-203
Subroutine CFALWD II-207
Subroutine LSCRN II-211
Subroutine RSCRN II-219
Subroutine QKINR II-223
Subroutine PDIST (NTYPE, LOGV, PRES, DIST) . . II-237

Subroutine PROMPT II-241
Subroutine RADKL II-249
Subroutine TRDCL II-259
Subroutine RDCNTY II-265

Volume III

DATA FILES III-1
A. Introduction III-1
B. File and Documentation Formats III-2
1. File Formats III-2
2. Documentation Formats III-4
C. Standard Format Record Definitions III-14
D. Standard Format File Descriptions III-26
E. Standard Format File Run Identifications . III-56
F. Standard Format File Data Tapes III-109
G. Standard Format File Card Decks III-143

ABSTRACT

This paper is a documentation of computer materials developed by the Institute for Defense Analyses (IDA) for use by the Defense Civil Preparedness Agency (DCPA). All IDA physical data processing materials (IBM cards, magnetic tape, computer printouts) have been surveyed and catalogued. All computer programs are written in FORTRAN (a general knowledge of this language is assumed in the detailed descriptions contained herein). Computer programs considered useful by IDA have been included and documented. A group of general purpose subprograms are described, along with their interfaces with the using programs. Data file formats also have been developed, along with programs for managing these files. Such programs and resulting files are described in detail.

PrecedING Page BLANK - NOT FILMed

Volume I

Chapter 1

I. INTRODUCTION

For over a decade the Institute for Defense Analyses has been conducting studies for the Defense Civil Preparedness Agency (or the Office of Civil Defense). Many of these studies have involved the development and use of computer programs. A legacy of such efforts is a substantial body of data processing materials collected over a number of years. During the last year IDA has engaged in an effort to identify these materials, to discard that no longer useful, to modify the remainder when appropriate to reflect present or possible future needs, and to document the resultant product. This paper presents the documentation.

Much of the material identified was considered no longer useful and was discarded. For example, data files based on the 1960 census were considered no longer useful and mostly discarded. (A few exceptions were made since the geographical basis of the 1970 census files is different than the National Shelter Survey, which used the 1960 Census RSAC codes.) In July 1970 IDA switched from a Control Data Corporation 1604 computer to a Control Data Corporation 6400 computer, with the result that many of the 1604 programs were unusable on the 6400 without extensive revision. The general criterion for selecting computer programs for retention was that the effort in understanding, modifying, and adapting an old program for possible future use be less than that of writing a new program. The general criteria for retaining old data files is that (1) the source of such data files (audit trail) is sufficiently well documented that the information is not unwittingly misused, and (2) new data files are not readily available covering the same subject but with more recent information. The result of applying these criteria resulted in many files discarded, with remaining files in a variety of formats, and led to the conclusion that the prime data effort should be in developing methodology for retaining data files, and put the most essential files in the format of this methodology.

It seems apparent that as this documentation becomes too voluminous that its utility is correspondingly decreased. Accordingly a conscious effort was made to only include in the documentation information which might be of substantial help.[1] Many previous IDA reports have contained descriptions and documentation of computer programs on data files. In such cases a reference to this information is given, and the information is not repeated here. The number of computer listings was kept to the minimum consistent with adequate documentation of pertinent materials.

For many years IDA has had a frequent and fruitful relationship with the DCPA Computer Center. A number of major programs have been originally developed at IDA but subsequently used (often with substantial modifications and improvements) at the DCPACC. Where such programs are maintained and available at the DCPACC, the documentation in this note of such programs will be restricted to that which might supplement that information already available at DCPACC. It is thus hoped that this note might be of help to the DCPACC when they use IDA originated programs by attempting to

[1] Material not specifically described is still available at IDA for use. In particular, catalogues and files of data tapes, IBM card decks, and computer produced listings of programs and results of runs are available in the Civil Defense Data Processing Library.

clarify obscure points, without needlessly replicating information already available.

A codification of many existing subprograms at IDA was conducted in 1972 under the acronym "NEVUNS." This material is the basis of Chapter 3 on subprograms. It was the hope at that time, and still appears as an appropriate goal, that many major programs could be rapidly constructed by assembling and selecting from a well tested and documented set of subprograms, each of which accomplishes a specific task. Many of these subprograms have, in fact, been utilized in programs developed since then and have resulted in substantial savings in effort. It is strongly hoped that in future activities this type of capability can be continued and expanded.

In the environment of development of programs intended to be run on more than one computer, a conscious effort should be continuously made to write code which is as machine independent as possible. It is thus hoped that much of the data processing material presented here will still be useful for advanced computer systems, and will not have to be discarded when the original machines they were developed on are discarded. In particular, the subprograms could be readily translatable to new machines as long as a reasonably complete version of FORTRAN is retained as a machine capability.

Chapter II

COMPUTER PROGRAMS

This chapter describes IDA computer programs. The selection of programs to be included followed the criteria of Chapter I. Most of the programs not included are either of a very temporary nature to test the correctness of a particular subprogram, are programs to accomplish small data file manipulation tasks, or are for obsolete machines. An exception to the exclusion of direct file manipulation routines are a set of special purpose, single use programs at the end of the chapter included to illustrate the development of standard files and document more fully the resultant files.

Where programs have been documented in previous references, the documentation is not given here. When programs are documented, the documentation consists of descriptive text followed by annotated listings. The form of the documentation varies considerably from program to program simply becuase different programs need different documentation techniques. In some cases, input requirements are transparent from the program listing, in others not. The program annotation describes the tasks performed by various sections and points out particular calculations where needed. When subprograms are NEVUNS Standard, the documentation is given in Chapter IV. The meanings of the critical variables are either described in the program header or in the preceding descriptive materials.

COMPUTER PROGRAM DESCRIPTION

NAME: DISCRN

SYNOPSIS: Fallout Dose Calculation

TYPE: Single Use

USE: To Calculate Hot Line Doses and Crosswind Standard Deviation in Fallout Risk Development

BACKGROUND: Developed to find First Approximations to WSEG-10

DESCRIPTION: Many Calls to WSEG-10 Model Followed by Printout of Appropriate Parameters

INPUT: In Program

OUTPUT: IDA Run #62

STORAGE: IDA Card Deck #40

DOCUMENTATION:

LANGUAGE/SYSTEM: RUN (FORTRAN)/6400 Scope

COMMENTS:

COMPUTER PROGRAM DESCRIPTION

NAME: SWOOSH

SYNOPSIS: WSEG-10 Fallout Dose for Many Conditions

TYPE: Single Use

USE: To Present Fallout Dose Values for Many Combinations of Yield, Wind Speed, Wind Shear, Downwind Distance and Crosswind Distance

BACKGROUND: Developed to Produce a Reference Catalog of Fallout Dose Values

DESCRIPTION: The WSEG-10 Model is Repeatedly Called for a Variety of Conditions Controlled by Do Loop. Dose Rates at Various Times and Crosswise Distances to Various Dose Levels Also Given

INPUT:
Parameters in Program

OUTPUT: IDA Listing #59-62

STORAGE: IDA Card Deck #208

DOCUMENTATION:

LANGUAGE/SYSTEM: Run (FORTRAN)/6400 SCOPE

COMMENTS: Copy of Catalog Supplied to DCPA

COMPUTER PROGRAM DESCRIPTION

NAME: DWNFIT

SYNOPSIS: Fit Fallout Crosswind Dose

TYPE: Single Use

USE: Determine shape of fallout doses crosswind as a function of downwind dose and attack pattern

BACKGROUND: Used as part of analysis in paper.

DESCRIPTION: Doses as a function of crosswind distance are computed with the WSEG-10 model and are presented for various parametric combinations of weapon yield, attack pattern, downwind distance, and wind

INPUT: Parameters in program.

OUTPUT: Printouts.

STORAGE: IDA Card Deck #50

DOCUMENTATION: IDA Paper P-1065, "Methodology of Fallout Risk Assessment," Leo A. Schmidt, January 1975

LANGUAGE/SYSTEM: FTN/6400 SCOPE

COMMENTS:

COMPUTER PROGRAM DESCRIPTION

NAME: WINDST

SYNOPSIS: Determine wind direction probabilities

TYPE: Single Use

USE: Used to compute probability wind has a direction θ as a function of θ with wind vector sum of a mean wind and circular Gaussian random component.

BACKGROUND: Written to obtain charts used in the risk paper.

DESCRIPTION: The probability is computed by using an expression which gives all possible random wind components with resultant direction θ and summing these probabilities.

INPUT: In program, cycles through various parameter values

OUTPUT: Printout

STORAGE: IDA Card Deck #35

DOCUMENTATION: IDA Paper P-1065, "Methodology of Fallout Risk Assessment," Leo A. Schmidt, January 1975

LANGUAGE/SYSTEM: FTN/6400 SCOPE

COMMENTS:

COMPUTER PROGRAM DESCRIPTION

NAME: LASH

SYNOPSIS: MONTE CARLO Fallout Dose Distributions

TYPE: Production

USE: Study Effect of Wind Statistics, Weapon Pattern on Fallout Doses

BACKGROUND: Developed as Exploratory Program in 1972
Used in Fallout Risk Studies--Bases of Other Applied Programs--RUBATO, TFPLOT COLPLT

DESCRIPTION: Has MONTE CARLO Simulation of Wind Statistics and Weapon Arrival to Compute Fallout Doses with WSEG-10 Fallout Model

INPUT: Parameter Cards-see IDA Card Decks 41, 97

OUTPUT: IDA Run 44, 51, 52, 74, 107

STORAGE: IDA Card Decks # 96, 97

DOCUMENTATION: IDA Paper P1065, "Methodology of Fallout-Risk Assessment" Leo A. Schmidt, Jr., Jan. 1975

LANGUAGE/SYSTEM: FTN/6400 SCOPE
FTN/3600 SCOPE
FTN/3200 SCOPE
6400 Version is ANSI Standard FORTRAN

COMMENTS:

3200 Version Experimental Interactive Program. Long Running on This Machine. For Other Machines Run Time Depends Upon Number of Trails, Number of Weapons, and Number of Monitor Points.

This program is described in technical details in a previous paper. This description will thus restrict itself to the computer implementation of the algorithms. The first part of the listing for the program RUBATO contains the definitions of program parameters.

The program reads a set of data cards in three subroutines PARIN, WPNIN, and TGTIN. The meaning of the input variables read in these routines will be described along with the implications on the resulting programs.

The following parameters are read in subroutine PARIN:

DMOST--For each monitor point the number of occurrences of doses in three types of intervals--even intervals, equal logarithmic intervals, and dose intervals specified in a data statement in the array BV are output. For the even intervals DMOST is the dose for the highest interval.

NDSEV--For the even interval doses, the variable gives the number of intervals.

SMLDOS--Since the dose distribution is highly skewed with a large number of small doses, dose statistics are presented for the full distribution and also only for those doses over the value SMLDOS.

WIND--Speed of the mean wind, mph.

SGWP--Standard deviation of random wind as a fraction of the mean wind speed.

SHR--Wind shear for WSEG-10 model use, mph/kft.

SGSHRP--Standard deviation of wind shear as a fraction of mean shear. The wind shear is assumed to be normally distributed.

MDCAL--Type of dose used. 0--Usual WSEG-10 biological dose, 1 is biological dose at 800 hrs., 2 is maximum biological dose.

NSAMP--Number of wind samples to be used

ISTRAT--If 1, do stratified wind samples. Take wind speed from each of ten equal probability intervals and wind direction from each of ten intervals. Number of samples should be an even multiple of 100.

MQCK--If 1, use rapid approximation to WSEG-10, if 0, use regular WSEG-10.

SEED--Random number generator seed. If this value is 0, use a computer clock reading to obtain a random seed.

IDEGI--If 1, input weapon coordinate by latitude and longitude. If 0, input target and weapon coordinate by miles, downwind and crosswind from some arbitrarily chosen origin.

WMNDR--Mean wind direction, degrees clockwise from north, only needed when IDEGI=1.

XCLAT--Center latitude weapon of target system. On input target location is converted to a rectangular coordinate system centered at XCLAT, YCLAT. This variable allows control of the origin of the rectangular system.

YCLON--Center longitude of target system.

IDEGC--As IDEGI but for target system. The following variables are input in the subroutine WPNIN for each weapon used.

YLDA(I)--Weapon yield in MT.

FISSA(I)--Weapon Fission Fraction.

HORA(I)--Weapon Height of Burst in feet.

TWPNA(I)--Time of weapon detonation, hrs. Not needed if MDCAL=0.

DCLA(I)--Weapon reliability

XWA(I)--Crosswind weapon distance

XWLAT(I)--Weapon latitude

YWA(I)--Crosswind weapon distance.

YWLON(I)--Weapon longitude.

If all yields are the same, the flag MONOY is set to 1, otherwise 0. If all yields are the same, yield and wind dependent WSEG-10 fallout calculations are done only once, if not, they are done for each weapon separately.

The following variables are read by the subroutine TGTIN.

For IDEGC=0

XTL(I)--downwind target location

YTL(I)--crosswind target location

For IDEGC=1

XCTLT(I)--target latitude

YCTLO(I)--target longitude

NAMEC(I,J,J=1,2)--target name

```
      PROGRAM LASH(INPUT,OUTPUT)

C     DOES A MONTE CARLO CALCULATION OF THE DISTRIBUTION OF DOSES FOR
C     CIRCULAR NORMAL RANDOM WIND VECTORS
C     L. SCHMIDT EXT. 346

      COMMON/WPNPR/ YLDA(200),FISSA(200),CEPA(200),HOBA(200),TWPNA(200),
     1 DELA(200), XWA(200), YWA(200), XWLAT(200),YWLON(200), IMAX,
     2 XCLAT,YCLON,ARRYA(200,40),ARRYT(200)
      COMMON/STAPR/       XLBV(51),NB(50,50)  ,NBL(50,50), NBE(101,50),
     1 BE(102),                    DMAX(50), SD(50), SDS(50),
     2  SDC(50), SDF(50),BV(51),NBOXA,NBOXL,NBOXE,DMOST,NDSEV
     3 ,DTOT,SMLDOS,SSD(50),SSDS(50),SSDC(50),SSDF(50), NSML(50)
      COMMON/IOPR/IN,NI,IBUG
      COMMON/TARPR/XTL(50),YTL(50),JTGTS,XCTLT(50),YCTLO(50),NAMEC(50,2)
      COMMON/WNDPR/WIND,SGWP,SHR,SGSHRP,SGW,SIGSHB, SHRST,
     1MDCAL, NSAMP, MONOV, NRPT  ,XNSAMP ,ISTRAT,IDEGI,WMNDR,IDEGC
     2 ,SAL,CAL,V,MOCK
      COMMON/LAZY/SEED ,TTA,TTB,TTC,TTD,KTA,KTB,KTC,KTD

      IN = 5LINPUT
      NI = 6LOUTPUT

2     CONTINUE                                               Read Data and
      READ( IN,4)IOVER                                       Initialize
4     FORMAT(I10)
      CALL PARIN )
      CALL WPNIN } Read 1 set of data cards
      CALL TGTIN )
      CALL INIT
      NRPTS = NSAMP/100
      DELTH = 0.6283185 3                                    Read Data and
                                                             Initialize

30    CONTINUE                                               Step through set of
      IF (ISTRAT .EQ.1) GO TO 53                             [illegible]
      NRPT = NRPT + 1
      THETA = 6.28318530* RANF(0)
      TMZ = RANF(0)
      GO TO 54
53    CONTINUE
      POLD = 0.
      DO 31 IS= 1,10
      THOLD = 0.
      DO 32 JS= 1,10
      DO 33 KS= 1,NRPTS
      THETA = THOLD + DELTH*RANF(0)   Angle of Random Wind
      TMZ = POLD + 0.1*RANF(0)
54    CONTINUE
      VWR = SQRT( 2.*SGW* SGW *ALOG(1./(1.- TMZ)))  Speed of Random Wind
      CALL WNDCAL(VWR,THETA)
      DO 100 KT = 1,JTGTS                                    Step through monte points
      DTOT = 0.
      DO 40 K = 1,IMAX                                       Step through weapons
      COME = RANF(0)
      IF(DELA(K).LT. COME) GO TO 40  Sample for Weapon Reliability
```

Preceding Page Blank - NOT FILMED

```
      CALL DSTCL(K,KT)                        Find Doses + Add
40    CONTINUF                                                       Step through weapons 2
      CALL BOXFL(KT)                          Record Statistics
100   CONTINUE                                                       Step through samples

      IF(ISTRAT .EQ. 1) GO TO 55                                     Increment stratified
      IF( NRPT .LT. NSAMP) GO TO 30                                  Wind Collection
      GO TO 56
55    CONTINUF
33    CONTINUE
      THOLD = THOLU + DELTH
32    CONTINUF
      POLD = POLD + 5.1
31    CONTINUE                                                       Increment stratified
      NSAMP = 100*NHPTS                                              Wind Collection
      XNSAMP = NSAMP                                                 Step through set of Winds
56    CONTINUE

      CALL WRTLST                             Report Results

      IF(IOVER .LT. 0) GO TO 2                another set of data?
      STOP 6400
      END
                                                                     Main Program

      BLOCK DATA
      COMMON/STAPR/       XLBV(51),NR(50,50)  ,NBL(50,50), NBE(101,50),
     1 BE(102),                          DMAX(50), SD(50), SDS(50),
     2  SDC(50), SUF(50),BV(51),NBOXA,NBOXL,NBOXE,DMOST,NDSEV
     3 ,DTOT,SMLDOS,SSD(50),SSDS(50),SSDC(50),SSDF(50), NSML(50)

      DATA BV/
     20.0,10.,20.,50.,100.,  200., 300., 400., 500., 600., 700., 800.,
     3900., 1000., 2000., 3000., 4000., 6000., 8000., 10000., 20000.,
     4  40000., 60000., 80000., 100000., 200000. . 400000.,  1000000./
      END


      SUBROUTINE INIT


C     INITIALIZES THE CALCULATION


      COMMON/STAPR/       XLBV(51),NR(50,50)  ,NBL(50,50), NBE(101,50),
     1 BE(102),                          DMAX(50), SD(50), SDS(50),
     2  SDC(50), SUF(50),BV(51),NBOXA,NBOXL,NBOXE,DMOST,NDSEV
     3 ,DTOT,SMLDOS,SSD(50),SSDS(50),SSDC(50),SSDF(50), NSML(50)
      COMMON/IOPR/IN,NI,IBUG
      COMMON/WPNPR/ YLDA(200),FISSA(200),CEPA(200),HOBA(200),TWPNA(200),
     1 DELA(200),  XWA(200), YWA(200), XWLAT(200),YWLON(200), IMAX,
     2 XCLAT,YCLON,ARRYA(200,40),ARRYT(200)
      COMMON/WNDPR/WIND,SGWP,SHR,SGSHRP,SGW,SIGSHR, SHRST,
     1MOCAL, NSAMP, MONDY, NRPT   ,XNSAMP ,ISTRAT,IDEGI,WMNDR,IDEGC
     2 ,SAL,CAL,V,MOCK
      COMMON/TARPR/XTL(50),YTL(50),JTGTS,XCTLT(50),YCTLO(50),NAMEC(50,2)
      COMMON/LAZY/SEED ,TTA,TTB,TTC,TTD,KTA,KTB,KTC,KTD


      IBUG = 2
```

```
       IBUG = 1
       IBUG = 0
       NBOXA = 28
       NBOXL = 37
       NBOXE = NDSEV + 1
       XLBV(1) = 0.
       XLG = 0.
       DO 3 J = 2,NBOXL
       XLBV(J) = 10.**XLG
       XLG = XLG + 0.2
 3     CONTINUE
       DELD = DMOST/NDSEV
       DD = 0.
       DO 5 J = 1,NBOXE
       BE(J) = DD
       DD = DD + DELD
 5     CONTINUE
       BV(NBOXA + 1)           = 999999999.
       XLBV(NBOXL +1)          = 999999999.
       BE(NBOXE + 1)           = 999999999.


       DO 54 K = 1,JTGTS
       DO 51 J = 1,NBOXA
       NB(J,K) = 0
 51    CONTINUE
       DO 52 J = 1,NBOXL
       NBL(J,K) = 0
 52    CONTINUE
       DO 53 J = 1,NBOXE
       NBE(J,K)  = 0
 53    CONTINUE
       NB( NBOXA + 1,K) = 0
       NBL(NBOXL + 1,K) = 0
       NBE(NBOXE + 1,K) = 0
       DMAX(K) = 0.
       SD(K) = 0.
       SDS(K) = 0.
       SDC(K) = 0.
       SDF(K) = 0.
       SSD(K) = 0.
       SSDS(K) = 0.
       SSDC(K) = 0.
       SSDF(K) = 0.
       NSML(K) = 0
 54    CONTINUE
       NRPT = 0


       IF( SEED .NE. 0.) GO TO 32
       CALL TIME(CLTIM)
       CLTIM = ABS(CLTIM )
       CALL RANSET(CLTIM)
       GO TO 33
 32    CONTINUE
       CALL RANSET(SEED)
 33    CONTINUE
       IF(MONOY .EQ.1) GO TO 27
       DO 25 I = 1,IMAX
       IF(MOCK .EQ. 1) GO TO 35
```

Initialize Random number Generator

Yield Dependent [illegible]
Calculation

```
      CALL FALLY(YLDA(I), FISSA(I), HOBA(I), ARRYT)
      GO TO 37
35    CONTINUE
      CALLQFALLY(YLDA(I), FISSA(I), HOBA(I), ARRYT)
37    CONTINUE
      DO 26 J = 1,6
      ARRYA(I,J)  = ARRYT(J)
26    CONTINUE
25    CONTINUE
      GO TO 28
27    CONTINUE
      IF(MQCK .EQ. 1) GO TO 36
      CALL FALLY(YLDA(1), FISSA(1), HOBA(1), ARRYT)
      GO TO 38
36    CONTINUE
      CALLQFALLY(YLDA(1), FISSA(1), HOBA(1), ARRYT)
38    CONTINUE
      DO 29 I = 1,IMAX
      DO 23 J = 1,6
      ARRYA(I,J) = ARRYT(J)
23    CONTINUE
29    CONTINUE
28    CONTINUE  .
      RETURN
      END
```

yield dependent use
10 calculation

```
      SUBROUTINE PARIN

C     INPUTS PARAMETRIC VARIABLES INCLUDING WIND AND CONTROL VALUES

      COMMON/WNDPR/WIND,SGWP,SHR,SGSHRP,SGW,SIGSHR, SHRST,
     1MDCAL, NSAMP, MONOY, NRPT  ,XNSAMP ,ISTRAT,IDEGI,WMNDR,IDEGC
     2 ,SAL,CAL,V,MOCK
      COMMON/STAPR/      XLBV(51),NB(50,50)  ,NBL(50,50), NBE(101,50),
     1 BE(102),                   DMAX(50), SD(50), SDS(50),
     2  SDC(50), SDF(50),BV(51),NBOXA,NBOXL,NBOXE,DMOST,NDSEV
     3 ,DTOT,SMLDOS,SSD(50),SSDS(50),SSDC(50),SSDF(50), NSML(50)
      COMMON/WPNPR/ YLDA(200),FISSA(200),CEPA(200),HOBA(200),TWPNA(200),
     1 DELA(200),  XWA(200), YWA(200), XWLAT(200),YWLON(200), IMAX,
     2 XCLAT,YCLON,ARRYA(200,40),ARRYT(200)
      COMMON/TOPR/IN,NI,IRUG
      COMMON/LAZY/SEED ,TTA,TTB,TTC,TTD,KTA,KTB,KTC,KTD


      READ(IN,4) DMOST, NDSEV,  SMLDOS
4     FORMAT(  F10.U, I10, F10.0 )

      READ(IN,12) WIND, SGWP ,SHR ,SGSHRP
12    FORMAT(4F10.0)
      SGW = SGWP*WIND
      SIGSHR = SHR*SGSHRP
      SHRST = SHR
      WRITE(NI,77)
77    FORMAT(1H1)
      WRITE(NI,9) WIND,SGWP,SHR
9     FORMAT(1H0, *  FOR THIS RUN AVERAGE WIND SPEED = *,F10.4, * M.P.H.
     1 -- STANDARD DEVIATION = *,F10.4, * TIMES AVG-- WIND SHEAR =*,F10.
     24,* MPH/KFT.*    )
```

```
      WRITE(NI,7) SGSHRP
7     FORMAT(1H , *STD DEV OF SHEAR =* ,F10.5, * TIMES AVG SHR. IF STD D
     1EV IS ZERO DO NOT MONTE CARLO SHEAR*)

      READ(IN,13) MDCAL,NSAMP,  ISTRAT ,MQCK
13    FORMAT(4I10)
C     MDCAL 0 WSEG BIO DOSE ONLY, 1 NMCSSC BIO, 2 ALSO DMAX
C     MONOY =0 SEVERAL YIELDS, 1 ONLY ONE YIELD
      XNSAMP = NSAMP
      WRITE(NT,8) NSAMP
8     FORMAT(1H0, *  THE SAMPLE SIZE = *, I6)
      IF(MDCAL .NE. 0) GO TO  91
      WRITE(NT,95)
95    FORMAT(1H0,  *USE WSEG BIOLOGICAL DOSE*)
      GO TO 94
91    CONTINUF
      IF(MDCAL .NE.1) GO TO 92
      WRITE(NT,96)
96    FORMAT(1H0,  *USE NMCSSC BIOLOGICAL DOSE*)
      GO TO 94
92    CONTINUF
      IF(MDCAL .NE.2) GO TO 93
      WRITE(NT,97)
97    FORMAT(1H0,  * USE NMCSSC MAXIMUM DOSE*)
      GO TO 94
93    CONTINUF
      STOP666
94    CONTINUF
      IF(ISTRAT .EQ. 1) GO TO  15
      WRITE(NT,17)
17    FORMAT(1H0, * USE NON-STRATIFIED SAMPLES *)
      GO TO 16
15    CONTINUF
      WRITE(NT,18)
16    CONTINUF
18    FORMAT( 1H0, * USE SAMPLES STRATIFIED BY RADIUS AND ANGLE INTO 100
     1 EQUAL PROBABILITY AREAS* )
      IF(MQCK .EQ. 1) GO TO 41
      WRITE(NT ,43)
43    FORMAT (1H0, *  USE REGULAR WSEG 10 CALCULATION*)
      GO TO 42
41    CONTINUF
      WRITE(NT ,44)
44    FORMAT(1H0, *   USE QUICK APPROXIMATION TO REGULAR WSEG 10 CALCUL
     1ATION*)
42    CONTINUF

      READ( IN,32) SEED
32    FORMAT( F10.0)
      IF( SEED .NE. 0) GO TO 33
      WRITE(NT,35)
35    FORMAT(1H0, * USE CLOCK TO GIVE AUTOMATIC RANDOM NUMBER SEED*)
      GO TO 34
33    CONTINUF
      WRITE(NT,36) SEED
36    FORMAT( 1H0,*USE RANDOM NUMBER SEED OF*, F16.6)
34    CONTINUF

      READ(IN,21) IDEGI, WMNOR, XCLAT, YCLON,IDEGC
21    FORMAT( I10,3F10.0,  I10  )
      IF(IDEGI .NE.1) GO TO  25
      WRITE(NI,26)  XCLAT,YCLON
26    FORMAT( 1H0, *USE LAT LON FOR WEAPON INPUT, CENTER TARGET AT LAT =
     1 *, F10.2, * DEGREES AND LON = *, F10.2 ,* DEGREES* )
      WRITE(NI,27) WMNDR
27    FORMAT (1H ,* THE MEAN WIND DIRECTION IS *, F10.4,   * DEGREES CLO
     1CKWISE FROM THE NORTH * )
25    CONTINUE
      RETURN
      END


      SUBROUTINE WPNIN


C     READS WEAPON PARMETERS AND LOCATIONS


      COMMON/IOPR/IN,NI,IBUG
      COMMON/WPNPR/ YLDA(200),FISSA(200),CEPA(200),HOBA(200),TWPNA(200),
     1 DELA(200),  XWA(200), YWA(200), XWLAT(200),YWLON(200), IMAX,
     2 XCLAT,YCLON,ARRYA(200,40),ARRYT(200)
      COMMON/WNDPR/WIND,SGWP,SHR,SGSHRP,SGW,SIGSHR, SHRST,
     1MDCAL, NSAMP, MONOY, NRPT   ,XNSAMP ,ISTRAT,IDEGI,WMNDR,IDEGC
     2 ,SAL,CAL,V,MOCK

      WRITE(NI,74)
74    FORMAT(1H0)
      IF(IDEGT .EQ. 1 ) GO TO 31
      WRITE(NI,18)
18    FORMAT( 1H0, * INPUT WEAPON PARAMETERS *,//,10X,11H WPN. NO.  ,10H
     1  YIELD   ,10HFISS. FRAC  ,10H HT. BURST  ,10H TIME DET.  ,
     2  10HDEL. PROB.    ,
     3   10H DWN. LOC.  ,10H CRS. LOC.   , // )
      I = 1
      MONOY = 1
21    CONTINUE
      READ(IN,22) YLDA(I), FISSA(I), HOBA(I),TWPNA(I),DELA(I),
     1 XWA(I),YWA(I)
22    FORMAT( 7F 10.0)
      IF( YLDA(I) .LT. 0.) GO TO 24
      IF(I .NE.1) GO TO 14
      YIELDO = YLDA(I)
      GO TO 15
14    CONTINUE
      IF(YLDA(I)  .NE. YIELDO) MONOY = 0
15    CONTINUE
      WRITE(NI,19) I,YLDA(I) ,FISSA(I), HOBA(I), TWPNA(I), DELA(I),
     1 XWA(I),YWA(I)
19    FORMAT(1H ,10X, 1H(, I2, 1H),   3X,       7F10.3)
      I = I + 1
      GO TO 21
24    CONTINUE
      IMAX = I - 1
      RETURN


31    CONTINUE
C     WEAPON INPUT BY LATITUDE AND LONGITUDE
      CONV = 60.*6086.2/5280.
      TORAD = 3.14159265/180.
      WRITE(NI,32)
```

```
32    FORMAT( 1H0, * INPUT WEAPON PARAMETERS *,//,10X,11H WPN. NO.  ,10H
     1  YIELD    ,10HFISS. FRAC  ,10H HT. BURST  ,10H TIME DET.  ,
     2  10HDEL. PROB.    ,
     3   10H OWN. LOC.  ,10H CRS. LOC.   ,10H WPN. LAT. , 10H WPN. LON.
     4   ,//)
      I = 1
      MONOY = 1
      THT = (WMNOR - 270.)*3.14159265/180.
      SNTHT = SIN(THT)
      COSTHT = COS(THT)
35    CONTINUE
      READ(IN,22) YLDA(I), FISSA(I), HOBA(I),TWPNA(I),DELA(I),
     1 XWLAT(I), YWLON(I)
      IF( YLDA(I) .LT. 0.) GO TO 36
      IF(I .NE.1) GO TO 37
      YIELDO = YLDA(I)
      GO TO 38
37    CONTINUF
      IF(YLDA(I)  .NE. YIELDO) MONOY = 0
38    CONTINUE
      CFAC = 0.5*TORAD*(XWLAT(I) + XCLAT )
      CCFAC = COS(CFAC)
      DELLAT = ( XWLAT(I) - XCLAT ) * CONV
      DELLON =(YWLON(I) - YCLON )*CONV*CCFAC
      XWA(I)   = -DELLON*COSTHT - DELLAT* SNTHT
      YWA(I) = DELLAT*COSTHT - DELLON*SNTHT
      WRITE(NI,39) I,YLDA(I) ,FISSA(I), HOBA(I), TWPNA(I), DELA(I),
     1 XWA(I),YWA(I)   , XWLAT(I) ,YWLON(I)
39    FORMAT(1H ,10X, 1H(, I2, 1H),   3X,       9F10.3)
      I = I + 1
      GO TO 35
36    CONTINUF
      IMAX = I - 1
      RETURN
      END


      SUBROUTINE TGTIN


C     READS TARGET LOCATIONS


      COMMON/TARPP/XTL(50),YTL(50),JTGTS,XCTLT(50),YCTLO(50),NAMEC(50,2)
      COMMON/WNDPR/WIND,SGWP,SHR,SGSHRP,SGW,SIGSHP, SHRST,
     1MOCAL, NSAMP, MONOY, NRPT  ,XNSAMP ,ISTRAT,IDEGI,WMNOR,IDEGC
     2 ,SAL,CAL,V,MOCK
      COMMON/WPNPR/ YLDA(200),FISSA(200),CEPA(200),HOBA(200),TWPNA(200),
     1 DELA(200),  XWA(200), YWA(200), XWLAT(200),YWLON(200), IMAX,
     2 XCLAT,YCLON,ARRYA(200,40),ARRYT(200)
      COMMON/IOPR/IN,NI,IBUG

      IF( IDEGC .EQ. 1) GO TO 31
      J = 0
15    CONTINUF
      J = J + 1
      READ(IN,14) XTL(J),YTL(J)
14    FORMAT( 2F10.0)
      IF(XTL(J) .LT. 1000000.) GO TO 15
      JTGTS = J -1
      RETURN
```

```
31    CONTINUF
C     CITY INPUT BY LAT LON
      TORAD = 3.14159265/180.
      CONV = 60.*6080.2/5280.
      J = 0
      THT = (WMNUR - 270.)*3.14159265/180.
      SNTHT = SIN(THT)
      COSTHT = COS(THT)
21    CONTINUE
      J = J + 1
      READ(IN,41) XCTLT(J), YCTLO(J), NAMEC(J,1),NAMEC(J,2)
41    FORMAT( 2F10.0, 2A8 )
      IF(XCTLT(J) .GT. 1000000. ) GO TO 32
      CFAC = 0.5*TURAD*(XCTLT(J) + XCLAT)
      CCFAC = COS(CFAC)
      DELLAT = ( XCTLT(J) - XCLAT ) *CONV
      DELLON =(YCTLU(J) - YCLON ) *CONV* CCFAC
      XTL(J)  = -DELLON*COSTHT - DELLAT* SNTHT
      YTL(J) = DELLAT*COSTHT - DELLON*SNTHT
      GO TO 21
32    CONTINUF
      JTGTS = J - 1
      RETURN
      END


      SUBROUTINE WNDCAL (VWR,THETA)


C     FOR A GIVEN RANDOM WIND VECTOR CALCULATES RESULTANT WIND VECTOR


      COMMON/TOPR/IN,NI,IBUG
      COMMON/WNDPP/WIND,SGWP,SHR,SGSHRP,SGW,SIGSHR, SHRST,
     1MDCAL, NSAMP, MCNOY, NRPT  ,XNSAMP ,ISTRAT,IDEGI,WMNDR,IDEGC
     2 ,SAL,CAL,V,MUCK
      COMMON/WPNPR/ YLDA(200),FISSA(200),CEPA(200),HORA(200),TWPNA(200),
     1 DELA(200),  XWA(200), YWA(200), XWLAT(200),YWLON(200), IMAX,
     2 XCLAT,YCLON,ARRYA(200,40),ARRYT(200)


      VSQ = WIND*WIND + VWR*VWR - 2.*WIND*VWR* COS(THETA)
      V = SQRT(VSQ)          magnitude of Resultant Wind
      SAL = VWR* SIN(THETA)/V   } sin + cosine of resultant wind angle with downwind direction
      CAL = SQRT( 1. - SAL*SAL) }
      TEMP =VWR*VWR- (V*V + WIND*WIND)
      IF( TEMP .GT. 0.) CAL = -CAL
      IF(SGSHRP .EQ. 0) GO TO 42
      SUM = 0.                  }
      DO 43 J = 1,12            }  Gen's vertical shear from truncated
      SUM = SUM + RANF(0)       }  normal distribution of Wind Shear
43    CONTINUF                  }
      SHR = SHRST + SIGSHR*(SUM - 6.)/12.
      IF(SHR .LT. 0.) SHR = 0.
42    CONTINUF
      IF(IBUG .EQ. 0) GO TO 45
      WRITE(NT,46) THETA,VWR,V,SAL,CAL,    SHR                  }  [illegible]
46    FORMAT(1H0,*DEBUG FOR THETA VWR V SAL CAL SHR*,    6F12.4) }
45    CONTINUF
```

Fill arrays for WSEG 10 yields and Wind Sensitive Subroutines

```
          IF( MONOY .EQ. 1) GO TO 31
          DO  32 I = 1,IMAX
          DO 33 J = 1,6
          ARRYT(J) =  ARRYA(I,J)
   33     CONTINUE
          IF(MQCK .EQ. 1) GO TO 51
          CALL  FALLFW( V, SHR, ARRYT)
          GO TO 53
   51     CONTINUE
          CALL QFALFW ( V, SHR, ARRYT)
   53     CONTINUE
          DO  34 J = 1,19
          ARRYA(I,J) = ARRYT(J)
   34     CONTINUE
   32     CONTINUE
          GO TO 35
   31     CONTINUE
          DO 36 J = 1,6
          ARRYT(J) = ARRYA(1,J)
   36     CONTINUE
          IF(MQCK .EQ. 1) GO TO 52
          CALL FALLFW(V,SHR, ARRYT)
          GO TO 54
   52     CONTINUE
          CALL QFALFW ( V, SHR, ARRYT)
   54     CONTINUE
          DO 37 I = 1,IMAX
          DO 38 J = 1,19
          ARRYA(I,J) = ARRYT(J)
   38     CONTINUE
   37     CONTINUE
   35     CONTINUE
          RETURN
          END
```

Fill arrays for WSEG 10 yields and Wind Sensitive Subroutines

```
          SUBROUTINE DSTCL(K,KT)

C         COMPUTES DOWNWIND AND CROSSWIND DISTANCES AND CALL FALLOUT ROUTS.

          COMMON/WPNPR/ YLDA(200),FISSA(200),CEPA(200),HOBA(200),TWPNA(200),
         1 DELA(200),  XWA(200), YWA(200), XWLAT(200),YWLON(200), IMAX,
         2 XCLAT,YCLON,ARRYA(200,40),ARRYT(200)
          COMMON/TARPR/XTL(50),YTL(50),JTGTS,XCTLT(50),YCTLO(50),NAMEC(50,2)
          COMMON/WNDPR/WIND,SGWP,SHR,SGSHRP,SGW,SIGSHP, SHRST,
         1MDCAL, NSAMP, MONOY, NRPT  ,XNSAMP ,ISTRAT,IUEGI,WMNDR,IDEGC
         2 ,SAL,CAL,V,MQCK
          COMMON/STAPR/      XLBV(51),NB(50,50)  ,NBL(50,50), NBE(101,50),
         1 BE(102),                    DMAX(20), SD(50), SDS(50),
         2  SDC(50), SUF(50),BV(51),NBOXA,NBOXL,NBOXE.DMOST,NDSEV
         3 ,DTOT,SMLDOS,SSD(50),SSDS(50),SSDC(50),SSDF(50), NSML(50)
          COMMON/TOPR/IN,NI,IBUG

          XX = XTL(KT) - XWA(K)
          YY = YTL(KT) - YWA(K)
          XDW = XX*CAL + YY*SAL
          YCW =-XX*SAL + YY*CAL
```

Get Downwind and Crosswind Wind Distances

```
      IF(IBUG .EQ. 0) GO TO 55
      WRITE(NT,59) XTL(KT),YTL(KT),XWA(K),YWA(K),XX,YY,XDW,YCW
 56   FORMAT(1H ,* DEBUG FOR XTL(KT),YTL(KT),XWA(K),YWA(K) XX, YY*,
     1   6F12.4,/,* AND XDW,YDW *,              2F12,4)
 55   CONTINUE
      DO 41 J = 1,19
      ARRYT(J) = ARRYA(K,J)
 41   CONTINUE
      IF(MOCK .EQ. 1) GO TO 42
      CALL FALLDW( XDW,MDCAL, TWPNA(K), ARRYT)       } Get Fallout Dose
      CALL FALLCW(YCW,MDCAL,ARRYT)
      GO TO 43
 42   CONTINUE
      CALL QFALDW(XDW,ARRYT)
      CALL QFALCW( YCW,ARRYT)
      DTOT = DTOT + ARRYT(13)
      GO TO 49
 43   CONTINUE

      IF(MDCAL .NE.0) GO TO 47
      DTOT = DTOT + ARRYT(33)
      GO TO 49
 47   CONTINUE
      IF (MDCAL .NE. 1) GO TO 48
      DTOT = DTOT +  ARRYT(38)                       Sum Doses
      GO TO 49
 48   CONTINUE
      DTOT = DTOT + ARRYT(39)
 49   CONTINUE
      IF(IBUG .EQ. 0) GO TO 65
      WRITE( NI,96) ARRYT(33), ARRYT(38), ARRYT(39), DTOT
 96   FORMAT( 1H  ,*DEBUG FOR ARRY 33, 38, 39, AND DTOT*, 4F12.4)
      IF(IBUG .EQ.1) GO TO 57
      DO  58 JXX = 1,40
      WRITE(NT,59) JXX, ARRYT(JXX)
 59   FORMAT( 1H  , I10, F20.9)
 58   CONTINUE
 57   CONTINUE
 65   CONTINUE
      RETURN
      END


      SUBROUTINE BUXFL(KT)


C     DOES THE CHORE OF ACCUMULATING THE VARIOUS STATISTICS


      COMMON/STAPR/       XLBV(51),NB(50,50)  ,NBL(50,50), NBF(101,50),
     1 BE(102),                       DMAX(50), SD(50), SDS(50),
     2  SDC(50), SDF(50),BV(51),NBOXA,NBOXL,NBOXE,DMOST,NDSEV
     3 ,DTOT,SMLDOS,SSD(50),SSDS(50),SSDC(50),SSDF(50), NSML(50)
      COMMON/WPNPR/ YLDA(200),FISSA(200),CEPA(200),HOBA(200),TWPNA(200),
     1 DELA(200),  XWA(200), YWA(200), XWLAT(200),YWLON(200), IMAX,
     2 XCLAT,YCLON,ARRYA(200,40),ARRYT(200)
      COMMON/TARPR/XTL(50),YTL(50),JTGTS,XCTLT(50),YCTLO(50),NAMEC(50,2)
      COMMON/WNDPR/WTAD,SGWP,SHR,SGSHRP,SGW,SIGSHR, SHRST,
     1MDCAL, NSAMP, MCNOX, NRPT  ,XNSAMP ,ISTRAT,IUEGI,WMNDR,IDEGC
     2 ,SAL,CAL,V,MOCK
      COMMON/TOPR/IN,NI,IBUG
      SD(KT) = SD(KT) + DTOT
      TMA = DTOT*DTOT
      SDS(KT) = SDS(KT) + TMA
      TMB = TMA*DTOT
      SDC(KT) = SDC(KT) + TMB
      TMC = TMB*DTOT
      SDF(KT) = SDF(KT) + TMC
      IF(DTOT .LT. SMLDOS) GO TO 72
      SSD(KT) = SSU(KT) + DTOT
      SSDS(KT) = SSUS(KT) + TMA
      SSDC(KT) = SSDC(KT) + TMB
      SSDF(KT) = SSDF(KT) + TMC
```

Sums for moments of Dose Statistics

```
      GO TO 73
72    CONTINUF
      NSML(KT) = NSML(KT) + 1
73    CONTINUE
```

Total Dose statistics to several type of summaries

```
      IF(DMAX(KT) .LT. DTOT) DMAX(KT) = DTOT
      DO 61 J = 1,NBOXL
      IF(   DTOT .GE. XLBV(J) ) GO TO 61
      NBL(J-1,KT) = NBL(J-1,KT) + 1
      GO TO 62
61    CONTINUF
      NBL(NBOXL,KT) = NBL(NBOXL,KT) + 1
62    CONTINUE
      DO 63 J = 1, NBOXA
      IF( DTOT .GE. BV(J) ) GO TO 63
      NB(J-1,KT) = NB(J-1,KT ) + 1
      GO TO 64
63    CONTINUF
      NB(NBOXA,KT) = NB(NBOXA,KT) + 1
64    CONTINUF
      DO 65 J = 1,NBOXE
      IF(DTOT .GE. BE(J) )  GO TO 65
      NBE(J-1,KT) = NBE(J-1,KT) + 1
      GO TO 66
65    CONTINUF
      NBE(NBOXE,KT) = NBE(NBOXE,KT) + 1
66    CONTINUE
      RETURN
      END
```

Add dose statistics
to several types of summaries

```
      SUBROUTINE WRTLST

C     PERFORMS THE FINAL OUTPUT OF DOSE STATISTICS FOR EACH TARGET

      COMMON/STAPR/      XLBV(51),NB(50,50)  ,NBL(50,50), NBE(101,50),
     1 BE(102),                    DMAX(50), SD(50), SDS(50),
     2  SDC(50), SDF(50),BV(51),NBOXA,NBOXL,NBOXE,DMOST,NDSEV
     3 ,DTOT,SMLDOS,SSD(50),SSDS(50),SSDC(50),SSDF(50), NSML(50)
      COMMON/TARPR/XTL(50),YTL(50),JTGTS,XCTLT(50),YCTLO(50),NAMEC(50,2)
      COMMON/IOPR/IN,NI,IBUG
      COMMON/WNDPR/WIND,SGWP,SHR,SGSHRP,SGW,SIGSHR, SHRST,
     1MDCAL, NSAMP, MONOY, NRPT  ,XNSAMP ,ISTRAT,IDEGI,WMNDR,IDEGC
     2 ,SAL,CAL,V,MOCK

      DO 101 KT = 1,JTGTS
      NSSUM = 0
      NLSUM = 0
      NESUM = 0
      WRITE(NT, 77)
77    FORMAT(1H1)
      WRITE( NI,74)
74    FORMAT(1H0)
      IF(IDEGC .EQ. 1) GO TO  91
      WRITE(NT,75) KT, XTL(KT), YTL(KT)
75    FORMAT(1H0, *DISTIRBUTIONS FOR TARGET NO.*, I2, * LOCATED*, F10.2,
     1 * STATUTE MILES DOWN AND*, F10.2,  * STATUTE MILES ACROSS FROM TH
     2E CENTRAL AXIS*)
      GO TO 92
91    CONTINUE
      WRITE(NT,93 ) NAMEC(KT,1), NAMEC(KT,2), KT, XCTLT(KT),YCTLO(KT),
     1 XTL(KT),  YTL(KT)
93    FORMAT(1H0,  * DISTRIBUTIONS FOR THE TARGET *, 2A8, * NUMBERED *
     1 ,I3,/, * AT LATITUDE *  ,F10.4 ,* DEGREES NORTH AND LONGITUDE *,
     2 F10.4 , * DEGREES WEST*,/, *  LOCATED*, F10.2, * STATUTE MILES DO
     3WN AND *, F10.2,  * STATUTE MILES ACROSS FROM THE CENTRAL AXIS*)
92    CONTINUE
      WRITE(NT,78) DMAX(KT)
78    FORMAT( 1H0, * THE MAXIMUM DOSE =* F13.4   )
      XMN = SD(KT)/XNSAMP
      XMNS = XMN*XMN
      XNUS = SDS(KT)/XNSAMP
      VAR = XNUS - XMNS
      STD = SQRT(VAR)
      XNUC = SDC(KT)/XNSAMP
      SKW = (XNUC - 3.*XNUS*XMN + 2.*XMNS*XMN)/(2.*VAR*STD)
      XNUF = SDF(KT)/XNSAMP
      XKUR=( XNUF - 4.*XNUC*XMN + 6.*XNUS*XMNS - 3.*XMNS*XMNS)/(VAR*VAR)
      WRITE(NT,76)XMN,STD,SKW,XKUR
76    FORMAT(1H0,*DOSE STATISTICS -- MEAN = *, F10.4,* STANDARD DEVIATIO
     1N = *, F10.4,  * SKEWNESS =*,F10.4,* KURTOSIS =*,F10.4    )
      IF( NSML(KT) .GT. 0)  GO TO 41
      WRITE (NI,42) SMLDOS
42    FORMAT( 1H0, ,* NO DOSES LESS THAN *, F10.1 )
      GO TO 46
41    CONTINUE
      IF(NSML(KT) .GT. 1) GO TO 43
      TEMP = SD(KT) - SSD(KT)
      WRITE(NT,44) TEMP,SMLDOS
44    FORMAT( 1H0, * ONE DOSE OF*, F10.2 , * IS LESS THAN DOSE OF*,
     1  F10.2)
      GO TO 46
43    CONTINUE
      XNSML = NSML(KT)
      XNSML = XNSAMP - XNSML
      XMN =SSD(KT)/XNSML
      XMNS = XMN*XMN
      XNUS =SSDS(KT)/XNSML
      VAR = XNUS - XMNS
      STD = SQRT(VAR)
      XNUC =SSDC(KT)/XNSML
      SKW = (XNUC - 3.*XNUS*XMN + 2.*XMNS*XMN)/(2.*VAR*STD)
      XNUF =SSDF(KT)/XNSML
      XKUR=( XNUF - 4.*XNUC*XMN + 6.*XNUS*XMNS - 3.*XMNS*XMNS)/(VAR*VAR)
      WRITE(NT,47) NSML(KT),SMLDOS, XMN,STD,SKW,XKUR
47    FORMAT( 1H0, * DOSE STATISTICS EXCLUDING * ,I6  , * DOSES UNDER *,
     1 F8.0, * RADS --*,/, 10X, *MEAN = *,F10.4, * STANDARD DEVIATION =
      2 *, F10.4,  * SKEWNESS = *, F10.4, * KURTOSIS = * ,F10.4)
46     CONTINUF

       WRITE( NI,74)
       WRITE(NT, 71)
71     FORMAT( 1H0, * SPECIFIED BOXES*///10X, 6H NO.    10X, 10HMIN. VALU
      1E    ,10x,  10HMAX, VALUE    , 10X,    10H NUMBER    ,10X,
      2  10H CUML. NO.        )
       DO 72 J = 1,NBOXA
       JJ = J + 1
       NSSUM = NSSUM + NB(J,KT)
       WRITE(NT,73)  J , BV(J) , BV(JJ), NB(J ,KT)  ,NSSUM
73     FORMAT( 1H  , 10X, 1H( , I2, 1H), 10X, F10.0,10X, F10.0, 10X,
      1 I10  ,10X,I10 )
72     CONTINUF
       WRITE(NT,81)
81     FORMAT( 1H1, * EQUAL LOGARITHMIC INTERVALS *,////,
      1 10x, 6H  NO.    , 10X, 10HMIN. VALUE  , 12X. 10HMAX. VALUE,
      2  12X,  10H NUMBER     ,10X, 10H CUML. NO.        )
       DO 83 J = 1,NBOXL
       JJ = J + 1
       NLSUM = NLSUM + NBL(J,KT)
       WRITE(NT,84) J. XLBV(J) , XLBV(JJ), NBL(J ,KT) ,NLSUM
84     FORMAT( 1H   , 10X, 1H(, I2, 1H), 10X, F12.2. 10X, F12.2,10X,I10,
      1  ,10x, I10)
83     CONTINUF
       WRITE(NT,85)
85     FORMAT(1H1,//    ,* EQUAL INTERVALS*, ////
      1,  10X, 6H NO.   ,10X, 10HMIN. VALUE    ,10X. 10HMAX. VALUE     ,
      2  10X,  10H NUMBER      ,10X, 10H CUML. NO.   )
       DO 86 J = 1,NBOXE
       JJ = J + 1
       NESUM = NESUM + NBE(J,KT)
       WRITE(NT,87) J, BE(J),BE(JJ). NBE(J ,KT)   ,NESUM
87     FORMAT( 1H , 10X, 2H (, I2, 2H)   ,10X, F10.0,10X,  F10.0,10X, I10,
      1  10X, I10   )
86     CONTINUF

101    CONTINUE
       RETURN
       END
       SUBROUTINE QFALLY( YIELD,FISS.HOB,QARRY)


C      A RAPID VERSION OF THE WSEG 10 FALLOUT MODEL BASED UPON FITS
C      TO CALCULATED DOSES RESULTS ARE STORED IN THE ARRAY QARRY . THE
C      BASIC ARRAY VARIABLES ARE
C      1...YIELD DEPENDENT ALPHA FACTOR IN FD
C      2...YIELD DEPENDENT BETA FACTOR IN FD
C      3...A IN SIGC CALCULATION
C      4...B IN SIGC CALCULATION
C      5...FINAL HOB FACTOR
C      6...IF 0 VARIABLES IN INTERPOLATION RANGE, IF 1 THEY ARE NOT.
C      7...VALUE OF WIND
C      8...ALPHA FACTOR IN FD
C      9...SHEAR/WIND
C     10...2.SIGC.SIGC
C     11...FD
C     12...(1/(SQRT(2.PI)SIGC)) .FD
C     13...WSEG BIOLOGICAL DOSE
C     14 - 19 .. USED IN SMALL WIND OR DISTANCE CALCULATIONS
```

Writ number of doses in various dose groups

Writ number of doses in various dose groups.

NEUONS STANDARD

```
C     CALLING SEQUENCE IS SAME AS WITH THE REGULAR MODEL EXCEPT FOR
C     NMCSSC DOSE OPTION

C     YIELD DEPENDENT CALCULATIONS


      DIMENSION QARRY(1)
      QARRY(14) = YIELD
      QARRY(6) = 0.
      IF( YIELD .LT. 0.1 .OR. YIELD .GT. 30.) QARRY(6) = 1.
      XLNY = ALOG10(YIELD)
      TMP = XLNY*XLNY
      QARRY(1) = 5.495- 0.1099*XLNY + 0.018*TMP
      QARRY(2) =-0.0641 + 0.0139*XLNY - 0.0033*TMP
      QARRY(3) = 2. + 1.7309*XLNY + 1.2691*TMP
      QARRY(4) = 7.55 + 1.8714*XLNY - 0.3314*TMP
      QARRY(5) = FISS
      IF(HOB .GT. 0.) GO TO 5
      RETURN
5     CONTINUE
      XMHB=180.*(YIELD*1000.)**0.4
      IF(HOB.LE.XMHB) GO TO 10
      QARRY(5) = 0.
      RETURN
   10 CONTINUE
      TEMP=HOB/XMHB
      AF=0.5*(1.-TEMP)*(1.-TEMP)*(2.+TEMP)+0.001*TEMP
      QARRY(5) = FISS * AF
      RETURN
      END
```

NEVUNS STANDARD

```
      SUBROUTINE QFALFW( EFW,SC,QARRY)

C     WIND DEPENDENT CALCULATIONS

      DIMENSION QARRY(1)
      QARRY(7) = EFW
      IF(EFW.LT. 3.) GO TO 5
      QARRY(9) = SC/EFW
      QARRY(8) = QARRY(1) - 0.995*ALOG10(EFW)
      RETURN
5     CONTINUE
      QARRY(9) = SC
      QARRY(6) = 2.
      WHS = 1. - 0.5*EFW
      QARRY(15) = (4.545 - 0.0745*QARRY(14)) + (0.1222 + 0.0078*
     1 QARRY(14) )*WHS - (1.2222 + 0.0278*QARRY(14) ) *SC
      QARRY(16) = -0.36486 + 0.00316*QARRY(14)
      QARRY(17)=( 0.2444 - 0.02444*QARRY(14) -(0.3977+ 0.1323*QARRY(14))
     1 *WHS )*0.001
      QARRY(18) = 3.14 + 0.51*QARRY(14) - (0.33+ 0.03*QARRY(14))*EFW
     1 + (42.35+ (-19.0975 + 0.9225*QARRY(14))*EFW)*SC
     2 + (69. + (-27.35  + 1.15 *QARRY(14))*EFW)*SC*SC
      QARRY(19) = (3.411 - 0.039*QARRY(14) )*SC
      RETURN
      END
```

NEVUNS STANDARD

```
      SUBROUTINE  QFALDW(DWD,QARRY)
```

15

```
C     DOWNWIND DISTANCE DEPENDENT CALCULATIONS

      DIMENSION QARRY(1)
      IF( QARRY(7) .LT. 3.) GO TO 20
      XSCL = DWD/QARRY(7)
      SIGC = QARRY(3) + QARRY(4)*QARRY(9)*DWD
      IF(XSCL .LT . 1.) GO TO 5
      XLGFD = QARRY(8) + QARRY(2)*XSCL
      TUSE = 2.5066.SIGC
      IF(XSCL .GT. 15.6) GO TO 10
      TMP = XSCL - 15.6
      XLGFD = XLGFU + 0.0015*TMP*TMP
10    CONTINUF
      QARRY(10) = 2.*SIGC*SIGC
      IF( XLGFD .LT. -8.) XLGFD = -8.
      QARRY(11) = QARRY(5) * 10.**XLGFD*QARRY(14)
      QARRY(12) = QARRY(11)/TUSE
      RETURN
5     CONTINUE
      XLW = ALOG10(QARRY(7)/20.)
      DELTN = 3. + 5.6*XLW
      FACT = 2. - ALOG10(QARRY(14))
      DELTA = DELTN*FACT
      XLDMXN = 3.355 - 0.386.XLW - 0.275.QARRY(9)/QARRY(7)
      XLDMX = XLDMXN + 0.448*(FACT - 1.)
      DWDS = DWD*FACT
      IF ( DWDS .LT. DELTA) GO TO 6
      XLGFD = XLDMX
      TUSE = 1.
      GO TO 10
6     CONTINUE
      XLGFD = XLDMX- (DWDS-DELTA)*(DWDS-DELTA)*0.0169
      GO TO 10
20    CONTINUF
      IF( DWD .GT.0. )GO TO 21
      XLY = ALOG10(QARRY(14))
      XLDMX = 4.35 - 2.56*XLY       - 0.12*QARRY(7)-0.15*QARRY( 9)
      BOT = 67. + 257.*ALOG10(QARRY(14)   )
      DUSE = 0.5*QARRY(7) - DWD
      XLGFD = XLDMX _ DUSE*DUSE/BOT
      GO TO 22
21    CONTINUF
      XLGFD = QARRY(15) + QARRY(16)*DWD + QARRY(17)*DWD*DWD
      IF(QARRY(17) .LT. 1.E-8) GO TO 22
      BETA = -1.*QARRY(16)/(2.*QARRY(17))
      IF( DWD .GT. BETA) XLGFD = -8.
22    CONTINUF
      SIGC = QARRY(18) + QARRY(19)*DWD
      TUSF = 1.
      GO TO 10
      END

      SUBROUTINE QFALCW( CWD,QARRY)

C     CROSSWIND DISTANCE DEPENDENT CALCULATIONS

      DIMENSION QARRY(1)
      TMP = CWD.CWD/QARRY(10)
      IF(TMP .GT.10.) TMP = 10.
      QARRY(13) = QARRY(12)*EXP(-TMP)
```

NcVUns STANDARD

```
      RETURN
      END
         1  IOVER IF POS DO NOT REPEAT CALCULATION
50000.          50   10.      VALUES OF DMOST  NDS EV AND SMLDOS
20.        0.3       .2           0.      VALUES OF WIND SIGW SHR SIGSH
      0    10000             0            1   MDCAL NSAMP ISTRAT MUCK
 0.0     SEED IF 0 USE CLOCK
      0    270.      40.          90.              0 DEGW MNWND LATC LONC CITYDEG
 1.        1.        0.           0.           1.0        0.0          0.0  YEHTDXY
 -1.                                                                        END WP RD
 20.0      0.        XTAR YTAR
   40.0    0.        XTAR YTAR
60.0       0.        XTAR YTAR
 100.      0.        XTAR YTAR
150.0      0.        XTAR YTAR
   40.0   10.        XTAR YTAR
   40.0   20.        XTAR YTAR
   40.0   40.        XTAR YTAR
99999999.                END TARGET READ
```

SAMPLE DATA IN

COMPUTER PROGRAM DESCRIPTION

NAME: LASHCO

SYNOPSIS: Produces Grid Fallout Risk Output

TYPE: Semi-Production

USE: Version of Lash to Plot Contours on a Grid

BACKGROUND: Dervied (in 1973 & 1975) from Program LASHTX Which in Turn was Derived from LASH. Used for Contour Plots in Colorado and near San Antonio, Texas

DESCRIPTION: Modification of LASH to put Monitor Points on a Grid and Produce Magnetic Tapes

INPUT: LASH Data Plus Weapon Locations
Card Deck #115

OUTPUT: IDA Runs #53, 54, 78, 79.
Tape Input to Program COLPLT

STORAGE: IDA Card Deck 113

DOCUMENTATION:

LANGUAGE/SYSTEM: Run(FORTRAN)/6400 SCOPE

COMMENTS:

COMPUTER PROGRAM DESCRIPTION

NAME: COLPLT
SYNOPSIS: Plot Grid Fallout Risk Data with Calcomp Plotter
TYPE: Semi-Production
USE: Uses LASHCO Tape Output to Plot Data

BACKGROUND: Supplementary Program to Program LASHCO

DESCRIPTION: Produces Fallout Plots Using IDA Calcomp Plot Package

INPUT: LASHCO Tape

OUTPUT: Calcomp Plot Tape, IDA Listing #50, 53, 54

STORAGE: IDA Card Deck 116

DOCUMENTATION:

LANGUAGE/SYSTEM: Run(FORTRAN)/6400 SCOPE

COMMENTS:

COMPUTER PROGRAM DESCRIPTION

NAME: RUBATO

SYNOPSIS: Nationwide Fallout Risk Calculation

TYPE: Production

USE: Computes Fallout Risk Statistics Both for Exploratory Purposes and as Input to ADAGIO Evacuation Calculations

BACKGROUND: First Developed in October 1973 on a Crises Basis to Provide ADAGIO Input for CRP Calculations. Used LASH Program as Basis.

DESCRIPTION: MONTE CARLO Simulation of Wind Statistics. Uses County Population Centroids as Monitor Points. Several Bookkeeping Features Including Distributions of Casualties & Fatalities from Fallout.

INPUT:

OUTPUT: IDA Listings 37-44, 103,109

STORAGE: IDA Card Deck 92

DOCUMENTATION: IDA Paper P1065, "Methodology of Fallout-Risk Assessment" Leo A. Schmidt, Jr., Jan. 1975
Attached Description

LANGUAGE/SYSTEM: FTN/6400 SCOPE
FTN/3600 SCOPE

COMMENTS: CDC 3600 Version Used Extensively at DCPACC. This Version Somewhat Different than Current IDA Version Due to Different Modification Requirements.

The program RUBATO was derived from the program LASH and is identical in many of the basic portions. Many of the variables have the same names and meanings. The first portion of the program RUBATO contains definitions of all significant variables.

The program was structured so that it could be expanded in several directions--primarily a better handling of relations between WSEG-10 model and cluster models to give better geographic resolution of the results, the capability of handling more realistic wind streamlines, and the capability of appending a blast fatality calculation. Unfortunately, time was never available to include these various refinements. The program is thus, partially completed, but has been used extensively for production calculations. The version presented is the current IDA version. This differs slightly from the DCPACC version.

The program has separate parameter input with the subroutine INVAL and descriptions of parameters in the subroutine PAROUT, as compared to LASH where the two are combined in PARIN. The input variable meaning is clear from the variable descriptions in the start of the program and the program LASH, except for the following remarks.

The switch IWNDST is only implemented for options 0, common statistics, or 1, use of the nearest monitor point as input in the wind statistics input at the end of subroutine INVAL.

The switch IWLVSW is only implemented for option 1, linear winds relative to the target.

The switch ICLSA is only implemented for option 1, weapons in clusters. These three switches allow for the types of calculations described in P-1065.

Activating the switch IFRCRS allows fallout fatality statistics to be developed. With each wind sample, the numbers of fatalities at the monitor point is calculated and statistics concerning fatalities are accumulated. It should be emphasized that without a blast fatality model, the fallout fatality calculations may be biased. For attacks in urban areas, where blast fatalities may occur, the effects of wind variations are less severe than in rural areas, where no blast fatalities generally occur.

The basic control of the calculation is in the subroutine MONTE. In particular, this subroutine generates individual winds, does the weapon screening, and calls the fallout dose subroutines.

```
      PROGRAM RUBATO(INPUT,OUTPUT,PUNCH,TAPEA,TAPEB,TAPEC,TAPED)          /
C **********3600/6400 COMPATABILITY CHANGE **************
C     PROGRAM RUBATO

C     TEMPORARY VERSION OF OCT. 5, 1973.
C     SLIGHTLY REVISED IN SEPT. 1974

C     A PROGRAM WHICH CONSIDERS TARGETS ONE AT A TIME IN A VARIETY OF
C     WAYS AND THEN PRODUCES NATIONWIDE SUMMARIES.
C     DOES A MONTE CARLO CALCULATION OF THE DISTRIBUTION OF DOSES FOR
C     CIRCULAR NORMAL RANDOM WIND VECTORS


      DIMENSION ILAB(3)
      COMMON/RUNSW/ICTYCB,IBINT,ICLUST,ISINGW,LABEL(20,2),ISPWSW,ILGWSW,
     1IEVWSW,DMOST,NDSEV,IHSWSW,ILVSW,DOSELV(5),IPROSW,PROLV(3),ICATSW,
     2 MOSTP,MLVTP,MMPTP,ICLSA,JREST,JRESTN,JRESTA,JRESTD,JRESTE,JRESTF,
     3 IFRSK,IDMAS,NSAMP,ISTRAT,IWNDST,ISHRSW,IWLVSW,MUCK,ISEED,SEED,
     4 MSDSW,CSEED,IFRCAS,NPFLV,PFLVU(7),PFLVR(7),FDSEM,FDSESG,CDSEM,
     5 CDSESG,MDCAL,ICLYLD,ICLYLQ,FRPFU(7) ,FRPFR(7)
     6,ILSTCT, ISTIN(50),ICTIN(100),ISEAS,NSTIN,NCTIN

C     RUNSW CONTIANS CONTROL PARAMETERS DEFINED BY INPUT DATA

C     INPUT CONTROL PARAMETERS
C        ICTYCB-- USE NNNN COUNTY DATA, CB AND CT RECORDS FOR TARGETS.
C           1 IF NO EOF,S AFTER EACH STATE,2 IF THERE IS
C        IBINT-- 0 BCD INPUT OF TARGET DATA, 1 BINARY INPUT.
C        ICLUST-- EXERCISE CLUSTER WEAPON INPUT
C        ISINGW-- EXERCISE SINGLE WEAPON INPUT
C        ICLYLD-- IF 0 USE AVERAGE YIELD AS INPUT FOR USE IN CLWSEG,
C           IF 1 USE A SINGLE YIELD FOR ENTIRE CALCULATION, FIRST VALUE
C           INPUT, IF 2 SELECT FROM ONE OF SEVEN CLASSES AS STANDARD,
C        IF 3 SELECT  CLASSES AS WEAPONS ARE INPUT
C        ICLYLQ-- AS ICLYLD BUT FOR USE WITH QUICK WSEG ROUTINES
C        ILSTCT-- IF 1 INCLUDE STATES IN ISTIN,2 ONLT CTY,S IN ICTIN
C           PLUS ALL OF ANY STATES IN ISTIN
C        ISTIN(50)-- LIST OF STATES TO BE USED IN A2 FORMAT
C        NSTIN-- NUMBER OF STATES FOR USE
C        ICTIN(100)-- LIST OF STATES AND CNTYS. TO BE USED IN A2,A3
C        NCTIN-- NUMBER OF STATE COUNTY COMBINATIONS TO DO

C     OUTPUT CONTROL PARAMETERS
C        LABEL-- TWO  80 CHARACTER LINES CENTERED ON TITLE PAGE.
C        ISPWSW-- WRITE DOSE LEVEL SPECIFIED IN ARRAY DV.
C        ILGWSW-- WRITE DOSES AT LOGARITHMIC INTERVALS.
C        IEVWSW-- WRITE DOSES AT EVEN INTERVALS
C        DMOST-- DOSE LEVEL AT HIGHEST OF EVEN INTERVALS.
C        NDSEV-- NUMBER OF EVEN INTERVALS, DIMENSION LIMIT OF 500.
C        IHSSW-- WRITE HISTOGRAMS OF DOSE DISTRIBUTION.
C        ILSFSW-- IF ONE OUTPUT FATALITIES, CASUALTIES AND DOSES FOR
C           INDIVIDUAL SAMPLES
C        ILVSW-- WRITE PROBABILITY OF DOSE AT SPECIFIED LEVELS.
C        DOSELV(5)-- SPECIFIED LEVELS FOR DOSES
C        IPROSW-- WRITE DOSE AT SPECIFIED PROB LEVELS.10, 50, 90 IF 1
C        PROLV(5)-- IF IPROSW = 2 USE THESE INPUT PROBABILITY LEVELS
C        ICATSW-- WRITE IN FORMAT FOR CATAGORY LEVEL OUTPUT
C        MOSTP--WRITE TAPE ON MD WITH DOSE IN SPECIFIED RAD BOXES
C        MLVTP-- WRITE TAPE ME WITH NUMBER AT SPECIFIED PERCENTILE PTS.
C        MMPTP-- WRITE EACH DOSE COMPUTED ON NE
```

Preceding Page Blank - NOT FILMED

```
C        ISGN-- IF SENSE SWITCH 1 SET DUMP ON DUMP TAPE NG
C        JSGN-- IF SENSE SWITCH 2 SET RESTART FROM DUMP TAPE NG.
C        KSGN-- IF SENSE SWITCH 3 SET DISPLAY CURRENT COUNTY ON CONSOLE
C        JREST-- CONTROLS POSITIONS OF TAPES ON RESTART, 0 ALL TAPES
C           AT 0, 1 ADV. TO JRESIN,ETC.,2 ADV TO DUMP TAPE VALUE.
C           POSITIVE INPUT TAPE ONLY,NEG. INPUT AND DOSE TAPES
C           ADVANCE DOSE TAPES TO EOF IF JREST = 2
C           JRESTN-- NUMBER OF TARGET (NUTR) TO RESTART WITH NEW TAPES.
C        JRESTA-- RECORD COUNT ON NA
C        JRESTD-- RECORD COUNT ON ND
C        JRESTE-- RECORD COUNT ON NE
C        JRESTF-- RECORD COUNT ON NF

C     RUN CONTROL OPTIONS.
C        IFRSK-- PERFORM FALLOUT RISK CALCULATIONS.
C        IDMAS-- PERFORM DAMAGE ASSESSMENT CALCULATIONS.
C        NSAMP-- NUMBER OF SMAPLES FOR EACH TARGET
C        ISTRAT-- IF 0 USE REGULAR SAMPLING, IF 1 USE STRATIFIED
C           SAMPLING.
C        IWNDST-- 0 USE INPUT VALUES WMNDR, WIND, SGWP FOR ENTIRE
C           CALCULATION, IF 1 USE MONITOR DATA POINTS AT FLATMO, ETC.
C           FOR AVERAGE WIND DATA AT TARGETS. IF 2 USE ALGORITHM
C           TO COMPUTE VALUES AT EACH TARGET.
C        ISEAS-- SEASON FOR WIND STATISTICS, 1-WINTER, 2-SPRING,
C           3-SUMMER, 4-FALL, 5-AVERAGE ANNUAL.
C        ISHRSW-- IF 0 USE CONSTANT SHEAR, IF 1 USE 1/2 NORMAL
C           DISTRIBUTION WITH MEAN ZERO AND STD DEV = SHRST*SGSHRP,
C           IF 2 USE FULL NORMAL WITH MEAN SHRST AND STD. DEV. =
C           SHRST*SGSHRP.
C        IWLVSW-- IF 0 USE ALL HORIZONTAL WIND, IF 1 USE LINEAR WINDS
C           RELATIVE TO TARGET, IF 2 USE PARABOLIC WINDS, IF 3 USE WIND
C           COORDINATE SYSTEM. 2 AND 3 NOT IMPLEMENTED YET.
C        ICLSA-- IF 0 USE INDIVIDUAL WEAPONS IN ANALYSIS, IF 1 USE
C           ONLY WEAPONS IN CLUSTERS, IF 2 USE COMBINATION.
C        MQCK-- IF 0 USE REGULAR WSEG 10 CALCULATIONS, 2F 1 USE QUICK
C           APPROXIMAIION
C        MOCAL--  IF 0 USE WSEG BIO DOSE ONLY, 1 NMCSSC BIO, IF 2
C           ALSO COMPUTE D MAX. FOR REGULAR WSEG 10 FALLOUT CALC.
C        ISEED-- IF 0 USE CLOCK FOR SEED, IF 1 USE IMPUT VALUE FOR SEED
C           FOR RANDOM NUMBER GENERATOR.
C        SEED-- INPUT SEED IF ISEED = 1
C        MSDSW-- USE SAME SEED FOR EACH TARGET
C        CSEED-- VALUE OF SEED TO  USE FOR TARGETS IF MSDSW = 1
C        IFRCAS-- IF 1 DO FALLOUT RISK CASUALTY CALCULATIONS.
C        NPFLV-- NUMBER OF PROTECTION FACTOR LEVELS TO ASSUME
C        PFLVU(7)-- PF VALUES FOR URBANIZED AREAS.
C        PFLVR(7)-- PF VALUES FOR NON URBANIZED AREAS.
C        FRPFR-- FRACTION WITH RURAL PF LEVEL
C        FRPFU-- FRACTION WITH URBAN PF LEVEL
C        FDSEM-- MEAN FATALITY FALLOUT DOSE
C        FDSESG-- STD DEV. OF MEAN FATALITY FALLOUT DOSE.
C        CDSEM-- MEAN INJURY FALLOUT DOSE.
C        CDSESG-- STD. DEV. ON MEAN INJURY FALLOUT DOSE.


      COMMON/RUNPR/ XNSAMP,NRPTS,NSMPCT,ILAST,ISTNO,ISTLO,JSTCD,JSTNM,
     1 ISTC(51),ISTNM(51),ICARRY(100),DSEPLT,IDSEPT,XPLT,YPLT
     2,DSEPLA,DSEPLB,DSEPLC,ISWING,JSWING,FRFATU,FRCASU,FRFATR,FRCASR

C     RUNPR CARRIES VARIOUS PARAMETERS NEEDED NOT INPUT DATA.
```

```
C         XNSAMP-- FLOATING POINT NUMBER OF SAMPLES.
C         NHPTS-- NUMBER OF TIMES AT EACH LEVEL IN STRATIFIED SAMPLES.
C         NSMPCT-- CURRENT NUMBER OF TRIALS IN NONE STRATIFIED SAMPLES.
C         ILAST-- FLAG TO INDICATE LAST TARGET HAS BEEN PROCESSED.
C         ISTNO-- NUMBER IN ORDER OF CURRENT STATE BEING PROCESSED.
C         ISTLU-- NUMBER IN LOOKUP ARRAY OF CURRENT STATE
C         JSTCD-- CODE NUMBER OF CURRENT STATE FROM LOOKUP ARRAY
C         JSTNM-- NAME OF CURRENT STATE FROM LOOKUP ARRAY.
C         ISTC(51)-- LOOKUP ARRAY OF STATE CODES IN 70 CENSUS.
C         ISTNM(51)-- LOOKUP ARRAY OF STATE NAMES.
C         ICARRY(100)-- POINTERS GIVEING ORDER IN WHICH STATES ARE
C            PROCESSED, VALUE IN ITH. ENTRY IS PLACE IN  ISTNM OF ITH ST.
C         DSEPLT-- VALUE OF DOSE TO USE FOR MAP PLOTTING
C         IDSEPT-- INDEX OF DOSE LEVEL TO USE FOR MAP PLOTTING.
C         XPLT-- X VALUE IN INCHES FROM MAP CENTER.
C         YPLT-- Y VALUE IN INCHES FROM MAP CENTER.
C         FRFAT-- FRACTION WITH FATALITIES FOR CURRENT DOSE
C         FRCAS-- FRACTION WITH CASUALTIES WITH CURRENT DOSE


      COMMON/CLSPR/ YLDC(500),FISSC(500),XWC(500),YWC(500),SIGXC(500),
     1 SIGYC(500),NCLS,SARY(500,5),YLDI(7),YLDM(6),QARYL(7,19),
     2ARRY(7,40)

C      CLSPR USED TO STORE DATA RELATED TO WEAPON CLUSTERS

C         FISSC-- FISSION FRACTION FOR CLUSTER, = TOTAL CLUSTER
C            FISSION YIELD / CLUSTER YIELD VALUE USED
C         YLDC--WEAPON YIELD,MT. THIS IS THE AVERAGE WEAPON YIELD USED
C            FOR A TYPICAL YIELD VALUE IN WEAPON EFFECTS CALCULATIONS.
C         XWC-- EAST WEST COORDINATE OF CENTER OF CLUSTER
C         YWC-- NORTH SOUTH COORDINATE OF CENTER OF CLUSTER
C         SIGXC-- EAST WEST STANDARD DEVIATION OF CLUSTER.
C         SIGYC-- NORTH SOUTH STANDARD DEVIATION OF CLUSTER
C         NCLS-- NUMBER OF CLUSTERS
C         SARY-- USED TO STORE YIELD DEPENDENT VALUES FOR CLUSTERED
C            WSEG - 10 FALLOUT MODEL
C         YLDI-- MEAN VALUE OF YIELD FOR GROUPED YIELD CALCULATIONS.
C         YLDM-- SEPERATING VALUES FOR YIELD INTERVALS
C         QARYL-- TO STORE YIELD DEPENDENT VALUE OF YIELD INTERVALS
C            USED IN QUICK WSEG 10 FALLOUT CALCULATIONS.


      COMMON/TARPR/ NAMETR(10),TRPT,TRPU,TRPR,TRPUA,IRSTCO,IRCOCO,
     1IRUAC,IREAC,ISMSAC,NUTR,TRCGLA,TRCGLO,YIL,XIL,SIGTRH,SIGTRL,
     2ALTR,NFLCS,NMCOS,NTT(52),FOTT(52),PTTT(52),PRTT(52),PATT(52)

C      TARPER USED FOR TARGET DATA

C         NAMETR-- TARGET NAME ARRAY, NORMAL USE IS 3A8,A6
C         TRPT-- TOTAL TARGET POPULATION.
C         TRPR-- RURAL TARGET POPULATION.
C         TRPU-- URBAN TARGET POPULATION.
C         TRPUA-- URBANIZED AREA TARGET POPULATION.
C         IRSTCO-- STATE CODE FROM 70 CENSUS, A2
C         IRCOCO-- COUNTY CODE FROM 70 CENSUS, A3
C         IRUAC-- URBANIZED AREA CODE, A4
C         IREAC-- ECOMOMIC AREA CODE' A5
C         ISMSAC-- SMSA CODE, A4
C         NUTR-- NUMBER OF CURRENT TARGET IN A SEQUENTIAL COUNT.
C         TRCGLA-- LATITUDE OF TARGET POPULATION CENTROID.
```

```
C         TRCGLO-- LONGITUDE OF TARGET CENTROID.
C         YTL-- TARGET CENTROID IN NORTH SOUTH COORDINATES.
C         XTL-- TARGET CENTROID IN EAST WEST COORDINATES.
C         SIGTRB-- STANDARD DEVIATION OF TARGET POPULATION ALONG BIG AXIS
C         SIGTRL-- STD. DEV. OF TARGET POPULATION ALONG SMALL AXIS.
C         ALTR-- ROTATION OF TARGET MAJOR (LARGE) AXIS IN DEGREES
C            CLOCKWISE FROM THE NORTH.
C         NPLCS-- NUMBER OF PLACES IN TARGET AREA
C         NMCDS-- NUMBER OF MINOR CIVIL DIVISIONS IN TARGET AREA.
C         NTT-- TOTAL NUMBER OF COUNTIES IN A STATE
C         PUTT-- TOTAL URBAN POPULATION IN A STATE.
C         PTTT-- TOTAL POPULATION IN A STATE
C         PRTT-- TOTAL RURAL POPULATION IN A STATE.
C         PATT- TOTAL URBANIZED AREA POPULATION IN A STATE.

      COMMON/WNDPR/WIND,V,SGWP,SGW,SHRST,SHR,SGSHRP,SGSHR,WMNDR,ALWND,
     1 CAL,SAL, FLATMO(100),FLONMO(100),DEGMO(100),SPDMO(100),VSMO(100),
     2 NUMO

C      WINDPR CONTAINS WIND RELATED PARAMETERS.

C         WIND-- CURRENT VALUE OF MEAN WIND, MPH.
C         V-- CURRENT VALUE OF SAMPLE WIND VELOCITY
C         SGWP-- WIND VECTOR STANDARD DEVIATION AS FRACTION OF MEAN WIND
C         SGW-- VECTOR STANDARD DEVIATION OF WIND. MPH.
C         SHRST-- MEAN VALUE OF WIND SHEAR MPH/KFT.
C         SHR-- CURRENT VALUE OF WIND SHEAR FOR THIS SAMPLE
C         SGSHRP-- STANDARD DEVIATION OF WIND SHEAR AS A FRACTION OF MEAN
C         SGSHR-- STANDARD DEVIATION OF WIND SHEAR.
C         WMNDR-- MEAN WIND DIRECTION IN DEGREES CLOCKWISE FROM THE NO.
C            I.E. A VALUE OF 270 MEANS A WIND BLOWING FROM WEST TO EAST
C         ALWND-- VALUE OF WIND DIRECTION FOR THIS SAMPLE.
C         CAL-- COSINE OF ALWND
C         SAL-- SINE OF ALWND.
C         FLATMO-- THESE ARRAYS REFER TO WIND MONITORING STATION VALUES
C            USED TO GIVE WIND STATISTICS DATA VARYING WITH LOCATION.
C            LATITUDE OF MONITORING STATION
C         FLONMO-- LONGITUDE OF STATION
C         DEGMO-- ANGLE OF MEAN WIND AT STATION IN DEGREES CLOCKWISE
C            FROM NORTH, AS WITH WMNDR
C         SPDMO-- MEAN WIND SPEED AT MONITORING STATION
C         VSMO-- VECTOR WIND STANDARD DEVIATION AS A FRACTION OF MEAN
C            WIND SPEED.
C         NUMO-- NUMBER OF MONITORING STATIONS.

      COMMON/STAPR/DTOT,DMAX,BV(51),BE(502),XLBV(51),NB(51),NBE(501),
     1NBL(50),NBOXA,NBOXL,NBOXE,SMLDOS,SD,SDS,SDC,SDF,SSD,SSDS,SSDC,SSDF
     2,NSML,NBXST(50,52),FBXST(100,52),CBXST(100,52),PROFT(50),PROCS(50)
     3,SFT,SFTS,SFTC,SFTF, SCA, SCAS,SCAC,SCAF,TPOPLV(50,52)
     4,DSEIND(100)

C      STAPR USED TO CARRY STATISTICS ON MONTE CARLO TRIALS.

C         NBXST-- NUMBER OF TRIALS IN SPECIFIED BOXES BY STATE
C         DTOT-- CURRENT VALUE OF TOTAL DOSE
C         DMAX-- CURRENT VALUE OF MAXIMUM DOSE.
C         BV(51)-- DOSE LEVEL CUTOFFS FOR SPECIFIED INTERVALS.
C         BE(502)-- DOSE LEVEL CUTOFFS FOR EVEN INTERVALS.
C         XLBV(51)-- DOSE LEVEL CUTOFFS FOR LOGARITHMIC INTERVALS
```

5

```
C          NB(51)-- TRIAL COUNT FOR SPECIFIED INTERVALS.
C          NBE(501)-- TRIAL COUNTS FOR EVEN INTERVALS.
C          NBL(50)-- TRIAL COUNTS FOR LOGARITHMIC INTERVALS.
C          NBOXA-- NUMBER OF BOXES FOR SPECIFIED INTERVALS.
C          NBOXL-- NUMBER OF BOXES FOR LOGARITHMIC INTERVALS.
C          NBOXE-- NUMBER OF BOXES FOR EVEN INTERVALS.
C          SMLDOS-- CUTOFF FOR STATISTICS WITHOUT DOSES UNDER SMLDOS
C          SD-- FOR THIS TARGET SUM OF DTOT
C          SDS-- SUM OF DTOT*DTOT
C          SDC-- SUM OF DTOT*DTOT*DTOT
C          SDF-- SUM OF DTOT*DTOT*DTOT*DTOT
C          SSD--  FOR DOSES OVER SMLDOS, SUM OF DTOT
C          SSDS-- FOR DOSES OVER SMLDOS, SUM OF DTOT*DTOT
C          SSDC-- FOR DOSES OVER SMLDOS, SUM OF DTOT*DTOT*DTOT
C          SSDF-- FOR DOSES OVER SMLDOS, SUM OF DTOT*DTOT*DTOT*DTOT
C          NSML-- NUMBER OF TRIALS LESS THAN SMLDOS FOR THIS TARGET.
C          NBXST-- USED FOR STATE ENTRIES IN SPECIFIED BOXES
C          FBXST-- USED FOR FRACTION FATALITIES FOR STATES.
C          CBXST-- USED FOR FRACTION CASUALTIES FOR STATES.
C          PROFT-- PROBABILITY OF A FATALITY AT A GIVEN DOSE LEVEL.
C          PROCS-- PROBABILITY OF A CASUALTY AT A GIVEN DOSE LEVEL.
C          SFT-- FOR EACH TRIAL SUM OF NUMBER OF FATALITIES.
C          SFTS-- SFT*SFT
C          SFTC-- SFT*SFT*SFT
C          SFTF-- SFT*SFT*SFT*SFT
C          SCA-- FOR EACH TRIAL SUM OF NUMBER OF CASUALTIES.
C          SCAS--SCA*SCA
C          SCAC-- SCA*SCA*SCA
C          SCAF-- SCA*SCA*SCA*SCA
C          TPOPLV-- TOTAL POPULATION AT SPECIFIED FATALITY LEVELS.
C          DSEIND(100)-- INDEX TO ORDERED LIST OF DOSES

      COMMON/IOPR/ MP,MQ,MS,NA,NC,ND,NE,NF,NG,NH,NI,NJ,NK,NL,NM

C     IOPR CONTAINS INPUT OUTPUT MEDIUM DEFINITIONS.

C        MP-- STANDARD INPUT - CARD READER.
C        MQ-- STANDARD OUTPUT - PRINTER.
C        MS-- OUTPUT OF BCD PUNCHED CARDS.
C        NA-- TARGET INPUT DATA.
C        NC-- WEAPON CLUSTER INPUT.
C        ND-- OUTPUT OF TARGET TRIAL NUMBERS IN SPECIFIED BOXES BY MDSTP
C        NE--OUTPUT OF DATA AT PERCENTILE LEVELS BY MLVTP.
C        NF-- OUTPUT OF INDIVIDUAL DOSES BY MMPTP
C        NG-- DUMP TAPE BY ISGN AND JSGN

      COMMON/LAZY/TTA,TTB,TTC,TTD,TTE,KTA,KTB,KTC,KTD,KTE,KTF,KTG,KTH

C     LAZY USED FOR TEMPORARY VARIABLE TRANSMITTAL.

C*****TEMPORARY*****
C        KTA USED TO TRANSMIT ILSFSW

      COMMON/LAZYA/TTAA(200),KTAA(200)

C     LAZYA USED FOR TEMPORARY ARRAY TRANSMITTAL.
```

```
      DATA  (BV(1) = 0.0),(BV(2) = 100.),(BV(3) = 200.),(BV(4) = 450.),
     1 (BV(5) = 1000.),(BV(6) = 2000.),(BV(7) = 4000.),(BV(8) = 10000.),
     2  (BV(9) = 30000.)
      DATA (YLDM(1) = 1.4),(YLDM(2) = 2.5) ,(YLDM(3) = 4.),(YLDM(4) =
     1  7.),( YLDM(5) = 14.),(YLDM(6) = 25.)
      DATA (YLDI(1) = 1.),(YLDI(2) = 2.),(YLDI(3) = 3.),(YLDI(4) = 5.),
     1(YLDI( 5) = 10.),(YLDI(6) = 20.),(YLDI(7) = 30.)
      DATA (( ISTC(J),J=1,51)=2H 1,2H 2,2H 4,2H 5,2H 6,2H 8,2H 9,2H10,2H
     111,2H12,2H13,2H15,2H16,2H17,2H18,2H19,2H20,2H21,2H22,2H23,2H24,2H2
     25,2H26,2H27,2H28,2H29,2H30,2H31,2H32,2H33,2H34,2H35,2H36,2H37,2H38
     3,2H39,2H40,2H41,2H42,2H44,2H45,2H46,2H47,2H48,2H49,2H50,2H51,2H53,
     42H54,2H55,2H56)
      DATA ((ISTNM(J),J=1,51)=2HAL,2HAK,2HAZ,2HAR,2HCA,2HCO,2HCT,2HDE,2H
     1DC,2HFL,2HGA,2HHI,2HID,2HIL,2HIN,2HIA,2HKS,2HKY,2HLA,2HME,2HMD,2HM
     2A,2HMI,2HMN,2HMS,2HMO,2HMT,2HNB,2HNV,2HNH,2HNJ,2HNM,2HNY,2HNC,2HND
     3,2HOH,2HOK,2HOR,2HPA,2HRI,2HSC,2HSD,2HTN,2HTX,2HUT,2HVT,2HVA,2HWA,
     42HWV,2HWI,2HWY)



C     CURRENTLY NC IS FROM CARDS
C **********36.0/6400 COMPATABILITY CHANGE **************
C     6400 CARDS
      MP = 5LINPUT
      MQ = 6LOUTPUT
      MS = 5LPUNCH
      NA = 5LTAPEA
      NC = 5LINPUT
      ND = 5LTAPEC
      NE = 5LTAPEB
      NG = 5LTAPED
C     3600 CARDS
C     MP=60
C     MQ=61
C     NA=20
C     NC=60
C     NE=30
C     ND=40
C     NG = 35

      ISTNO = 0
      NUTR = 0
      IOSTCD = 2H

C **********3600/6400 COMPATABILITY CHANGE **************
      CALL SSWTCH(2,JSGN)
C     IF(SENSE SWITCH 2) 8,9
      IF(JSGN .EQ. 1) GO TO 8
      GO TO 9
8     CONTINUE
C     RESTART FROM DUMP TAPE. REPOSITION PREVIOUS OUTPUT TAPES BY END
C     OF FILE MARKS. ASSUME PRINT OUTPUT FROM BEFORE SAVED ELSEWHERE.
      READ(MP,35) JREST,JRESTN,JRESTA,JRESTD,JRESTE,JRESTF
35    FORMAT(/,6I10)
      INOUT = 3
      CALL IDUMP(INOUT,IOSTCD,ISTLU)
      WRITE(MQ,11)
```

Restart
only tested on 6400.

Restart

```
        CALL INVAL
        CALL PAROUT
        GO TO 10
9       CONTINUE

        CALL INVAL
        CALL PAROUT
        IF(IFRSK .NE.1) GO TO 100
        IF(ICLUST .NE.1) GO TO 10
        CALL INIT
        CALL CLSIN
10      CONTINUE
        CALL TGTIN              Read a new monitor Point
C       **********************3600/6400 COMPATIBILITY CHANGE***********
        CALL SSWTCH(3,KSGN)                                    Consol Display
        IF(KSGN .EQ.1) GO TO 31
C       IF(SENSE SWITCH 3) 31,32
        GO TO 32
31      CONTINUE
        ITM = 9HST CTY =
        ENCODE(14,34, ILAB)ITM,IRSTCO,IRCOCO
34      FORMAT(A9,A2,A3)
        ILAB(3) = 0
        CALL REMARK(ILAB)
32      CONTINUE
        IF(ILAST .NE.0) GO TO 20
C ***********3600/6400 COMPATABILITY CHANGE **************
C       IF(SENSE SWITCH 1) 25,24                               Dump
        CALL SSWTCH(1,ISGN)
        IF(ISGN .EQ. 1) GO TO 25
        GO TO 24
25      CONTINUE
        INOUT = 1
        CALL IDUMP(INOUT,IOSTCO,ISTLU)
        GO TO 20                                               Dump
   24   CONTINUE
        NUTR = NUTR + 1                                        State typ bookkeping
        IF( IRSTCO .EQ.IOSTCD ) GO TO  12
        IOSTCD = IRSTCO
        ISTNO = ISTNO + 1
        IF(ISTNO.GT.51) GO TO 20
        DO 17 J = 1,51
        JJ = J
        IF(IRSTCO .EQ. ISTC(J) ) GO TO 18
17      CONTINUE
        ISTLU = 1
        GO TO 19
18      CONTINUE
        ISTLU = JJ
19      CONTINUE
        ICARRY(ISTNO) = ISTLU
        JSTCD = ISTC(ISTLU)
        JSTNM = ISTNM(ISTLU)
        WRITE(MG,11)
11      FORMAT(1H1)
        WRITE(MG,21)  ISTNM(ISTLU ),ISTC(ISTLU )
21      FORMAT(//////, 20X, *RESULTS FOR THE STATE OF *,  A2,
      1 * WITH CODE *, A2,////)
        ISWING = 1
```

```
12     CONTINUE
       NTT(ISTNO) = NTT(ISTNO) + 1
       PUTT(ISTNO) = PUTT(ISTNO) + TRPU
       PTTT(ISTNO) = PTTT(ISTNO) + TRPT
       PRTT(ISTNO) = PRTT(ISTNO) + TRPR
       PATT(ISTNO) = PATT(ISTNO) + TRPUA                        Stat Tape Bookkeep
       CALL PREPAR
       CALL MONTE
       CALL WRTONE
       IF(MLVTP .NE.1) GO TO 16
       WRITE(NE,15) TRCGLA,TRCGLO,DSEPLA,DSEPLB,DSEPLC,IRSTCO,IRCOCO
15     FORMAT(F10.4,5X,F10.4,5X,3F10.4,5X,A3,1X,A2)
16     CONTINUE
       GO TO 10

20     CONTINUE                                               Final Summary
       CALL WRTLST                                            + Close file
       IF( MDSTP .NE.1) GO TO 22
       ENDFILE ND
       REWIND ND
22     CONTINUE
       IF (MLVTP .NE.1) GO TO 26
       ENDFILE NE
       REWIND NE
26     CONTINUE
       IF(MMPTP .NE.1) GO TO 27
       ENDFILE NF
       REWIND NF
27     CONTINUE
100    CONTINUE
       STOP 6400                                              Final Summary
       END

       SUBROUTINE INIT

C      INITIALIZES THE CALCULATION

       COMMON/RUNSW/ICTYCB,IHINT,ICLUST,ISINGW,LABEL(20,2),ISPWSW,ILGWSW,
      1IEVWSW,CMOST,NDSEV,IHSWSW,ILVSW,DOSELV(5),IPROSW,PROLV(3),ICATSW,
      2 MDSTP,MLVTP,MMPTP,ICLSA,JREST,JRESTN,JRESTA,JRESTD,JRESTE,JRESTF,
      3 IFRSK,IDMAS,NSAMP,ISTRAT,IWNDST,ISHRSW,IWLVSW,MOCK,ISEED,SEED,
      4 MSDSW,CSEED,IFRCAS,NPFLV,PFLVU(7),PFLVR(7),FDSEM,FDSESG,CDSEM,
      5 CDSESG,MDCAL,ICLYLD,ICLYLG,FRPFU(7) ,FRPFR(7)
      6,ILSTCT, ISTIN(50),ICTIN(100),ISEAS,NSTIN,NCTIN
       COMMON/RUNPR/ XRSAMP,NRPTS,NSMPCT,ILAST,ISTNO,ISTLO,JSTCD,JSTNM,
      1 ISTC(51),ISTNM(51),ICARRY(100),DSEPLT,IDSEPT,XPLT,YPLT
      2,DSEPLA,DSEPLB,DSEPLC,ISWING,JSWING,FRFATU,FRCASU,FRFATR,FRCASR
       COMMON/CLSPP/ YLDC(500),FISSC(500),XWC(500),YWC(500),SIGXC(500),
      1 SIGYC(500),NCLS,SARY(500,5),YLDI(7),YLDM(6),QARYL(7,19),
      2ARRY(7,40)
       COMMON/TARPR/ NAMETR(10),TRPT,TRPU,TRPR,TRPUA,IRSTCO,IRCUCO,
      1IRUAC,IREAC,ISMSAC,NUTR,TRCGLA,TRCGLO,YTL,XTL,SIGTRB,SIGTRL,
      2ALTR,NPLCS,NMCDS,NTT(52),PUTT(52),PTTT(52),PRTT(52),PATT(52)
       COMMON/STAPR/DTCT,DMAX,HV(51),BE(502),XLBV(51),NB(51),NBE(501),
      1NPL(50),NBOXA,NBOXL,NBOXE,SMLDUS,SD,SUS,SDC,SDF,SSD,SSDS,SSDC,SSDF
      2,NSML,NBXST(50,52),FBXST(100,52),CBXST(100,52),PROFT(50),PROCS(50)
```

```
     3,SFT,SFIS,SFTC,SFTF, SCA, SCAS,SCAC,SCAF,TPOPLV(50,52)
     4,DSEINU(100)                                                    9
      COMMON/WWSEG/ YIELN,FISN,HOB,Z,SHZ,XDW,YCW,XTRA,JARRY(19)
      COMMON/LAZY/TTA,TTB,TTC,TTD,TTE,KTA,KTB,KTC,KTD,KTE,KTF,KTG,KTH
      COMMON/LAZYA/TTAA(200),KTAA(200)


      NBOXA = 9
      NBOXL = 37
      NBOXE = NDSEV + 1
      XLBV(1) = 0.
      XLG = 0.
      DO 3 J = 2,NBOXL
      XLBV(J) = 10.**XLG
      XLG = XLG + 0.2
3     CONTINUE
      XNDSEV = NDSEV
      DELD = DMOST/XNDSEV
      DD = 0.
      DO 5 J = 1,NBOXE
      BE(J) = DD
      DD = DD + DELD
5     CONTINUE
      BV(NBOXA + 1)                 = 999999999.
      XLBV(NBOXL + 1)               = 999999999.
      BE(NBOXE + 1)                 = 999999999.
      DO 11 I = 1,50
      DO 12 J = 1,52
      NBAST(I,J) = 0
      TPOPLV(I,J) = 0.
12    CONTINUE
11    CONTINUE
      DO 16 I = 1,100
      DO 17 J = 1,52
      FHAST(I,J) = 0.
      CHAST(I,J) = 0.
17    CONTINUE
16    CONTINUE

      DO 14 J = 1,52
      NTI(J) = 0
      PUTT(J) = 0.
      PTIT(J) = 0.
      PRIT(J) = 0.
      PAIT(J) = 0.
14    CONTINUE

      IF(ISEED .NE.0) GO TO 32
C **********3600/6400 COMPATABILITY CHANGE **************
      CALL TIME(CLTIM)
      CLTIM = ABS(CLTIM )
      CALL RANSET(CLTIM)
C     CLTIM = TIMEF(DUMMY)
C     CLTIM = ABS(CLTIM )
C     CALL RANFSET(CLTIM)
      GO TO 33
32    CONTINUE
C **********3600/6400 COMPATABILITY CHANGE **************
      CALL RANSET(SEED)
C     CALL RANFSET(SEED)
33    CONTINUE
```

```
          FISN = 1.
          HOB = 0.
          DO  38 I = 1,7
          YIELN = YLDI(I)                            Initialize Fallout Yield Calculation
          CALL QFALLY
          DO 39 J = 1,19
          QARYL(I,J) = QARRY(J)
39        CONTINUE
38        CONTINUE
          RETURN
          END

          SUBROUTINE INVAL

C         INPUTS PARAMETRIC VARIABLES INCLUDING WIND AND CONTROL VALUES

          COMMON/RUNSW/ICTYCB,IBINT,ICLUST,ISINGW,LABEL(20,2),ISPWSW,ILGWSW,
         1IEVWSW,DMOST,NDSEV,IHSWSW,ILVSW,DOSELV(5),IPROSW,PROLV(3),ICATSW,
         2 MUSTP,MLVTP,MMPTP,ICLSA,JREST,JRESTN,JRESTA,JRESTD,JRESTE,JRESTF,
         3 IFRSK,IDMAS,NSAMP,ISTRAT,IWNDST,ISHRSW,IWLVSW,MQCK,ISEED,SEED,
         4 MSDSW,CSEED,IFRCAS,NPFLV,PFLVU(7),PFLVR(7),FDSEM,FDSESG,CDSEM,
         5 CDSESG,MDCAL,ICLYLD,ICLYLW,FRPFU(7) ,FRPFR(7)
         6,ILSTCT, ISTIN(50),ICTIN(100),ISEAS,NSTIN,NCTIN
          COMMON/RUNPR/ XNSAMP,NRPIS,NSMPCT,ILAST,ISTNO,ISTLO,JSTCD,JSTNM,
         1 ISTC(51),ISTNM(51),ICARRY(100),DSEPLT,IDSEPT,XPLT,YPLT
         2,DSEPLA,DSEPLB,DSEPLC,ISWING,JSWING,FRFATU,FRCASU,FRFATR,FRCASR
          COMMON/WNDPR/WIND,V,SGWP,SGW,SHRSI,SHR,SGSHRP,SGSHR,WMNDR,ALWND,
         1 CAL,SAL, FLATMO(100),FLONMO(100),DEGMO(100),SPDMO(100),VSMO(100),
         2 NUMO
          COMMON/STAPR/DTOT,DMAX,BV(51),BE(502),XLBV(51),NB(51),NBE(501),
         1NBL(50),NBOXA,NBOXL,NBOXE,SMLDOS,SD,SDS,SDC,SDF,SSD,SSDS,SSDC,SSDF
         2,NSML,NBXST(50,52),FBXST(100,52),CBXST(100,52),PROFT(50),PROCS(50)
         3,SFT,SFTS,SFTC,SFTF, SCA, SCAS,SCAC,SCAF,TPOPLV(50,52)
         4,DSEIND(100)
          COMMON/IOPR/ MP,MQ,MS,NA,NC,ND,NE,NF,NG,NH,NI,NJ,NK,NL,NM
          COMMON/LAZY/TIA,TIB,TIC,TID,TTE,KIA,KIB,KIC,KTD,KTE,KTF,KIG,KTH
          COMMON/LAZYA/TTAA(200),KIAA(200)

          READ(MP,6)(LABEL(I,1),I= 1,20)
          READ(MP,6)(LABEL(I,2),I= 1,20)
6         FORMAT(20A4)
          READ(MP,1)  IFRSK,IDMAS
1         FORMAT(  /,2I10)

          READ(MP,21) ICTYCB,IBINT
21        FORMAT(/,2I10)
          READ(MP,21) ICLUST,ISINGW
          READ(MP,21) ICLYLD,ICLYLW
C
          READ(MP,41) ILSTCT
41        FORMAT(/,I10)
          IF(ILSTCT .EQ.0) GO TO 42

C         READ STATES TO USE FIRST, STOP ON SP,THEN COUNTIES STOP
C         ON SP. IF ILSTCT IS ONE ONLY ONE SP
          JJ = 0
```

```
44      CONTINUE
        JJ = JJ + 1
        READ(MP,43) ISTIN(JJ)
43      FORMAT(A2)
        IF( ISTIN(JJ) .NE. 2HSP ) GO TO 44
        NSTIN = JJ - 1
        IF(ILSTCT .EQ. 1) GO TO 42
        JJ = 0
46      CONTINUE
        JJ = JJ + 1
        READ( MP,45) ICTIN(JJ)
45      FORMAT(A5)
        IF( ICTIN(JJ) .NE. 5HSP     ) GO TO 46
        NCTIN = JJ - 1
42      CONTINUE

        READ(MP,26) ISPWSW,ILGWSW,IEVWSW,IHSWSW ,ILSFSW
26      FORMAT(/,5I10)
C*****TEMPORARY*****
        NTA = ILSFSW
        READ(MP,16) ICATSW
        READ(MP,4) DMOST, NOSEV,  SMLDOS
4       FORMAT(/,F10.0, I10, F10.0 )
        READ(MP,27)ILVSW,(DOSELV(I),I= 1,5)
27      FORMAT(/,I10,5F10.0)
        READ(MP,28)IPROSW,(PROLV(I),I= 1,3)
28      FORMAT(/,I10,3F10.0)
        READ(MP,34) MDSTP,MLVTP,MMPTP
34      FORMAT(/,3I10)
        READ(MP,22)JREST,JRESTN,JRESTA,JRESTD,JRESTE,JRESTF
22      FORMAT(/,6I10)

        READ(MP,21)NSAMP,ISTRAT
        XNSAMP = NSAMP
        NRPTS = NSAMP/100
        READ(MP,12)IWNDST,WIND,SGWP,WMNDR,ISEAS
12      FORMAT(/,I10,3F10.0,I10)

        READ(MP,14) ISHRSW,SHRST,SGSHRP
14      FORMAT( /,I10, 2F10.0)
        SGW = SGWP*WIND
        SGSHR = SHRST*S;SHRP
        READ(MP,16) IWLVSW
16      FORMAT(/,I10)
        READ(MP,16) ICLSA
        READ(MP,21)MUCK,MDCAL
        READ(MP,23) ISEED,SEED,MSDSW,CSEED
23      FORMAT(/,I10,F10.0,I10,F10.0)
        READ(MP,21) IFRCAS,NPFLV
        READ(MP,24) (PFLVU(I),I=1,7)
        READ(MP,24) (FRPFU(I),I=1,7)
        READ(MP,24) (PFLVR(I),I=1,7)
        READ(MP,24) (FRPFR(I),I=1,7)
24      FORMAT(/,7F10.0)
        READ(MP,35) FUSEM,FDSESG,CUSEM,CDSESG
35      FORMAT(/,4F10.0)

        IF(IWNDST .NE.1 ) GO TO 75
        NUMO = 0
        CO =  6080.2/5280.
```

Read Grid Wind Data

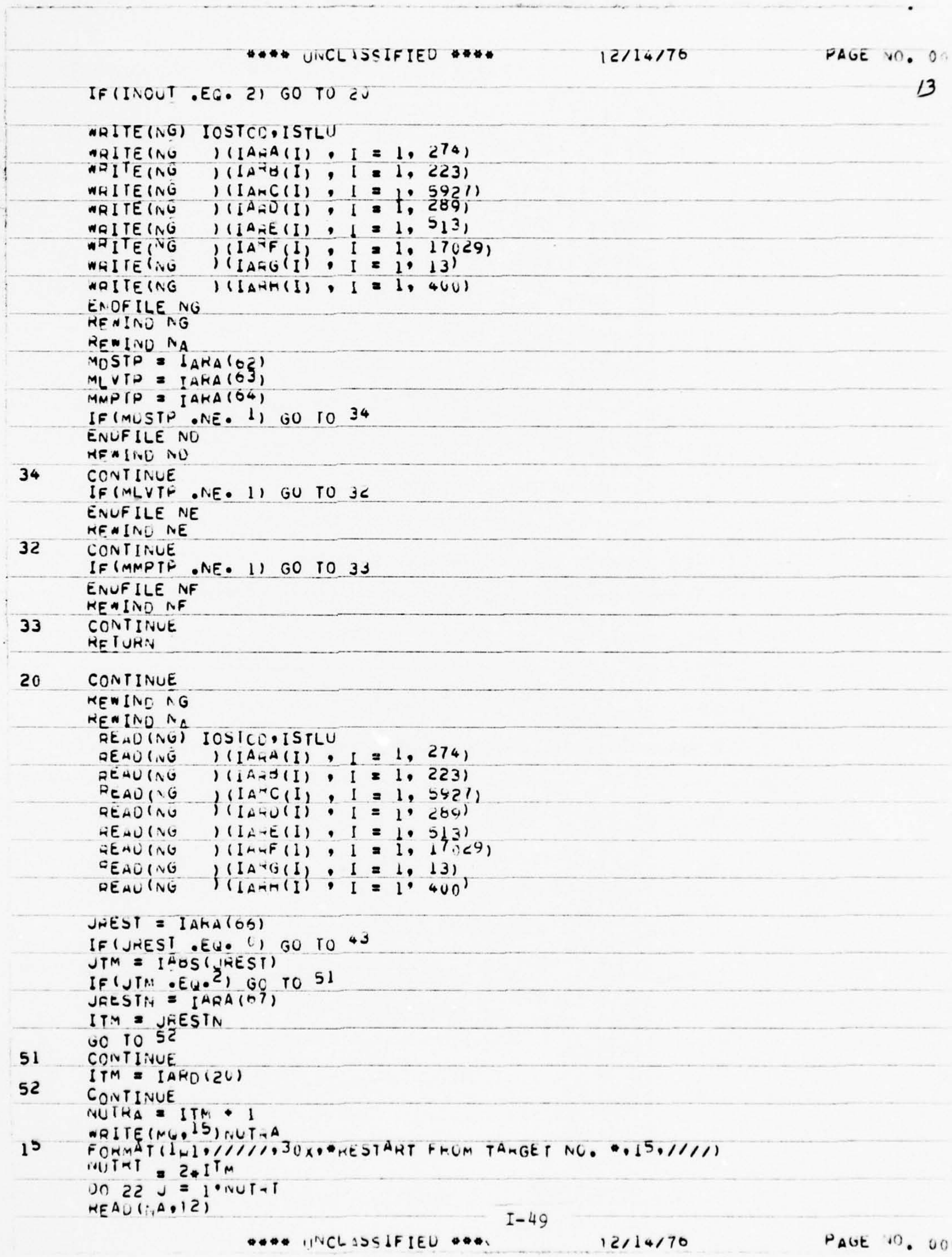
12

```
70      CONTINUE
        READ(MP,71) LAT,LON,IWD,IWS,IWSD,ISUD,ISUS,ISUSD,ISPD,ISPS,ISPSD,
      1 IFD,IFS,IFSD
71      FORMAT(I3,5X,  I3, 5X,I3,1X,I3,1X,I3,1X,I3,1X,I3,1X,I3,1X,I3,1X,
      1 I3,1X,I3,1X,I3,1X,I3,1X,I3)
        IF (LAT .EQ.0) GO TO 75
        NUMO = NUMO + 1
        FLATMO(NUMO) = LAT
        FLONMO(NUMO) = LON
        IF(ISEAS .NE.1) GO TO 81
        ZD = IWD
        ZS = IWS
        ZSD = IWSD
        GO TO 86
81      CONTINUE
        IF(ISEAS .NE.2) GO TO 82
        ZD = ISPD
        ZS = ISPS
        ZSD = ISPSD
        GO TO 86
82      CONTINUE
        IF(ISEAS .NE.3) GO TO 83
        ZD = ISUD
        ZS = ISUS
        ZSD = ISUSD
        GO TO 86
83      CONTINUE
        IF(ISEAS .NE.4) GO TO 84
        ZS = IFS
        ZD = IFD
        ZSD = IFSD
        GO TO 86
84      CONTINUE
        ZD = IWD + ISPD + ISUD + IFD
        ZD = 0.25*ZD
        ZS = IWS + ISPS + ISUS + IFS
        ZS = 0.25*ZS
        ZSD = IWSD + ISPSD + ISUSD + IFSD
        ZSD = 0.25*ZSD
86      CONTINUE
        DEGMO(NUMO) = ZD
        SPDMO(NUMO) = ZS*CO
        VSMO(NUMO) = ZSD*CO
        GO TO 70
75      CONTINUE
        RETURN
        END

        SUBROUTINE IDUMP(INOUT,IUSTCD,ISTLU)
C       INOUT = 1 WRITE ON DUMP TAPE
C       INOUT = 2 READ FROM DUMP TAPE
        COMMON/RUNSW/IARA(274)
        COMMON/RUNPR/IARB(223)
        COMMON/CLSPR/IARC(5927)
        COMMON/TARPR/IARD(299)
        COMMON/WNDPR/IARE(513)
        COMMON/STAPR/IARF(17029)
        COMMON/IOPR/ MP,MQ,MS,NA,NC,ND,NE,NF,NG,NH,NI,NJ,NK,NL,NM
        COMMON/LAZY/IARG(13)
        COMMON/LAZYA/IARH(400)
```

Grid Wind Data

not needed if General Checkpoint Restart Routine available



13

```
      IF(INOUT .EQ. 2) GO TO 20

      WRITE(NG) IOSTCD,ISTLU
      WRITE(NG   )(IARA(I) , I = 1, 274)
      WRITE(NG   )(IARB(I) , I = 1, 223)
      WRITE(NG   )(IARC(I) , I = 1, 5927)
      WRITE(NG   )(IARD(I) , I = 1, 289)
      WRITE(NG   )(IARE(I) , I = 1, 513)
      WRITE(NG   )(IARF(I) , I = 1, 17029)
      WRITE(NG   )(IARG(I) , I = 1, 13)
      WRITE(NG   )(IARH(I) , I = 1, 400)
      ENDFILE NG
      REWIND NG
      REWIND NA
      MDSTP = IARA(62)
      MLVTP = IARA(63)
      MMPTP = IARA(64)
      IF(MDSTP .NE. 1) GO TO 34
      ENDFILE ND
      REWIND ND
34    CONTINUE
      IF(MLVTP .NE. 1) GO TO 32
      ENDFILE NE
      REWIND NE
32    CONTINUE
      IF(MMPTP .NE. 1) GO TO 33
      ENDFILE NF
      REWIND NF
33    CONTINUE
      RETURN

20    CONTINUE
      REWIND NG
      REWIND NA
       READ(NG) IOSTCD,ISTLU
       READ(NG   )(IARA(I) , I = 1, 274)
       READ(NG   )(IARB(I) , I = 1, 223)
       READ(NG   )(IARC(I) , I = 1, 5927)
       READ(NG   )(IARD(I) , I = 1, 289)
       READ(NG   )(IARE(I) , I = 1, 513)
       READ(NG   )(IARF(I) , I = 1, 17029)
       READ(NG   )(IARG(I) , I = 1, 13)
       READ(NG   )(IARH(I) , I = 1, 400)

      JREST = IARA(66)
      IF(JREST .EQ. 0) GO TO 43
      JTM = IABS(JREST)
      IF(JTM .EQ.2) GO TO 51
      JRESTN = IARA(67)
      ITM = JRESTN
      GO TO 52
51    CONTINUE
      ITM = IARD(20)
52    CONTINUE
      NUTRA = ITM + 1
      WRITE(MG,15)NUTRA
15    FORMAT(1H1,/////,30X,*RESTART FROM TARGET NO. *,I5,////)
      NUTRT = 2*ITM
      DO 22 J = 1,NUTRT
      READ(NA,12)
```

]

```
12      FORMAT(1X)
22      CONTINUE
        IF(JREST .GT. 0) GO TO 43
        MDSTP = IARA(62)
        MLVTP = IARA(63)
        MMPTP = IARA(64)
        IF( JREST .NE. -2) GO TO 45
        JRESTA = IARA(68)
        JRESTD = IARA(69)
        JRESTE = IARA(70)
        JRESTF = IARA(71)
        IF (MDSTP .NE. 1) GO TO 46
        DO 47 JJ = 1,JRESTD
        READ(ND,12)
47      CONTINUE
46      CONTINUE
        IF(MLVTP .NE. 1) GO TO 48
        DO 49 JJ = 1,JRESTE
        READ(NE,12)
49      CONTINUE
48      CONTINUE
        IF( MMPTP .NE.1) GO TO 53
        NSAMP = IARB(1)
        KTM = NSAMP/10
        KKTM = KTM*10
        IF(KKTM .NE. NSAMP) KTM = KTM + 1
        KKTM = JRESTF*(KTM+1)
        DO 54 JJ = 1,KKTM
        READ(NF,12)
54      CONTINUE
53      CONTINUE
45      CONTINUE
        IF(MLVTP .NE. 1) GO TO 42
23      CONTINUE
        READ(NE,12)
C ***********3600/6400 COMPATABILITY CHANGE **************
        IFL = EOF(NE)
        IF(IFL .NE.  0) GO TO 24
C       IF(EOF,NE)24,25
25      CONTINUE
        GO TO 23
24      CONTINUE
        BACKSPACE NE
42      CONTINUE
        IF(MDSTP .NE. 1) GO TO 41
28      CONTINUE
        READ(ND,12)
C ***********3600/6400 COMPATABILITY CHANGE **************
        IFL = EOF(ND)
        IF(IFL .NE.  0) GO TO 26
C       IF(EOF,ND) 26,27
27      CONTINUE
        GO TO 28
26      CONTINUE
        BACKSPACE ND
41      CONTINUE
        IF(MMPTP .NE. 1) GO TO 43
29      CONTINUE
        READ(NF,12)
C ***********3600/6400 COMPATABILITY CHANGE **************
        IFL = EOF(NF)
```

15

```
      IF(IFL .NE. 0) GO TO 30
C     IF(EOF,NF)30,31
31    CONTINUE
      GO TO 29
30    CONTINUE
      BACKSPACE NF
43    CONTINUE
      RETURN
      END
```

End of Extract Routine

```
      SUBROUTINE PAROUT


      DIMENSION ITELL(2)
      DIMENSION JSEA(5)
      DIMENSION LINEP(130),CSPTX(50),CSPTY(50),SSPTA(50),SSPTL(50),
     1SALS(50),CALS(50),IPNT(50),KST(4)
      COMMON/RUNSW/ICTYCB,IBIN!,ICLUST,ISINGW,LABEL(20,2),ISP,SW,ILGWSW,
     1IEVWSW,UMOST,NDSEV,IHSWSW,ILVSW,DOSELV(5),IPROSW,PROLV(3),ICATSW,
     2 MDSTP,MLVTP,MMPTP,ICLSA,JREST,JRESTN,JRESTA,JRESTD,JRESTE,JRESTF,
     3 IFRSK,IDMAS,NSAMP,ISTRAT,IWNDST,ISHRSW,IWLVSW,MOCK,ISEED,SEED,
     4 MSDSW,CSEED,IFRCAS,NPFLV,PFLVU(7),PFLVR(7),FDSEM,FDSESG,CDSEM,
     5 CDSESG,MDCAL,ICLYLD,ICLYLW,FRPFU(7) ,FRPFR(7)
     6,ILSTCT, ISTIN(50),ICTIN(100),ISEAS,NSTIN,NCTIN
      COMMON/RUNPR/ XNSAMP,NRPTS,NSMPCT,ILAST,ISTNO,ISTLO,JSTCD,JSTNM,
     1 ISTC(51),ISTNM(51),ICARRY(100),DSEPLT,IDSEPT,XPLT,YPLT
     2,DSEPLA,DSEPLB,DSEPLC,ISWING,JSWING,FRFATU,FRCASU,FRFATR,FRCASR
      COMMON/WNDPR/WIND,V,SGWP,SGW,SHRST,SHR,SGSHRP,SGSHR,WMNDR,ALWND,
     1 CAL,SAL, FLATMO(100),FLONMO(100),DEGMO(100),SPDMO(100),VSMO(100),
     2 NUMO
      COMMON/IOPR/ MP,MQ,MS,NA,NC,ND,NE,NF,NG,NH,NI,NJ,NK,NL,NM
      COMMON/LAZY/TTA,TTB,TTC,TTD,TTE,KTA,KTB,KTC,KTD,KTE,KTF,KTG,KTH
      COMMON/LAZYA/TTAA(200),KTAA(200)

      DATA (ITELL(1) = 6HDO NOT   ),(ITELL(2) = 6H    DO   )
      DATA (JSEA(1) =6HWINTER ),(JSEA(2) = 6HSPRING   ),( JSEA(3) =
     1 6HSUMMER),( JSEA(4) = 6H FALL  ),(JSEA(5) = 6HANNUAL  )


      WRITE(MG,77)
C **********3600/6400 COMPATAHILITY CHANGE ***************
C     IZL = -1
      IZL = 0
      CALL TIME(CLTIM)
      CLTIM = ABS(CLTIM )
      CALL RANSET(CLTIM)
      TEM = RANF(IZL)
      LNCV = 29. + 10.*TEM
      TEM = RANF(IZL)
      NSPOT = 4. + 45.*TEM*TEM
      IJ = 0
      DO 14 I = 1,NSPOT
      TEM = RANF(IZL)
      CCPTX = 5. + 12 .*TEM
      TEM = RANF(IZL)
      CCPTY = 5. + 46.*TEM
      TEM = RANF(IZL)
      SCPTX = 2. + 14.*TEM
      TEM = RANF(IZL)
      SCPTY = SCPTX*(0.3 + 0.7*TEM)
      IF(I.EQ.I) GO TO 13
      DO 11  J = 1,IJ
      IF(CCPTX .GT.CSPTX(J))GO TO 12
      IF( CCPTX + SCPTX + 1..LT .CSPTX(J) - SSPTB(J)) GO TO 11
      GO TO 15
12    CONTINUE
      IF( CCPTX - SCPTX - 1. .GT.CSPTX(J) + SSPTB(J)) GO TO 11
15    CONTINUE
      IF(CCPTY .GT. CSPTY(J)) GO TO 16
      IF( CCPTY + SCPTY + 1. .GT. CSPTY(J) - SSPTL(J) ) GO TO 14
      GO TO 11
16    CONTINUE
      IF(CCPTY  - SCPTY - 1. .LT.CSPTY(J) + SSPTL(J)) GO TO 14
11    CONTINUE
13    CONTINUE
      IJ = IJ + 1
      CSPTX(IJ) = CCPTX
      CSPTY(IJ) = CCPTY
      SSPTB(IJ) = SCPTX
      SSPTL(IJ) = SCPTY
      TEM = RANF(IZL)
      ALPH = 3.14159265*TEM
      SALS(IJ)= SIN(ALPH)
      CALS(IJ)= COS(ALPH)
14    CONTINUE
      NSPOT = IJ
      TEM = RANF(IZL)
      ACN = 14. + 8.*TEM
      ACNS = ACN*ACN
      DO 21 K = 1,60
      XK = K
      IFL = 0
      DO 22 I = 1,NSPOT
      IF(ABS(CSPTY(I)-XK) .GT. SSPTB(I))GO TO 22
      IFL = IFL + 1
      IPNT(IFL) = I
22    CONTINUE
      DO 26 J = 1,129
      XJ = J
      SUM = 0.
      IF(IFL .EQ. 0) GO TO 28
      DO 27 I = 1,IFL
      M = IPNT(I)
      IF(ABS(CSPTX(M) -XJ).GT. SSPTB(M) ) GO TO 27
      DX = CSPTX(M) - XJ
      DY = CSPTY(M) - XK
      DXP = DX*CALS(M) + DY*SALS(M)
      DYP =  -DX*SALS(M) + DY*CALS(M)
      TMB = SSPTB(M)*SSPTB(M) +0.001
      TML = SSPTL(M)*SSPTL(M) + 0.001
      ALS = DXP*DXP/TMB + DYP*DYP/TML
      ALS = 1. - ALS
      IF(ALS .LE. 0.) GO TO 27
      SUM = SUM + ALS
27    CONTINUE
      IF(SUM .GT.0.1) GO TO 29
      LINEP(J) = 1H
      GO TO 26
```

```
29      CONTINUE
        TEM = RANF(IZL)
        VAL = 10.*SUM + 2.*(TEM -0.5)
        IF(VAL .GT.1.) GO TO 31
28      CONTINUE
        LINEP(J) = 1H
        GO TO 26
31      CONTINUE
        IF(VAL .GT.3.) GO TO 32
        LINEP(J) = 1H.
        GO TO 26
32      CONTINUE
        IF(VAL .GT.5.) GO TO 33
        LINEP(J) = 1H+
        GO TO 26
33      CONTINUE
        IF(VAL .GT.7.) GO TO 34
        LINEP(J) = 1HO
        GO TO 26
34      CONTINUE
        IF(VAL .GT.9.) GO TO 35
        LINEP(J) = 1HX
        GO TO 26
35      CONTINUE
        LINEP(J) = 1H*
26      CONTINUE
        ITM = LNCV - 1
        IF(K.LT.ITM) GO TO 40
        ITM = LNCV + 22
        IF(K.GE. ITM) GO TO 40
        YY = LNCV + 11 - K
        DO 41 I = 30,100
        XX = I-65
        ALS = XX*XX/ACNS + YY*YY/36.
        IF(ALS .GE.1.) GO TO 42
        LINEP(I) = 1H
        GOTO 41
42      CONTINUE
        IF(ALS .LE.2.77) GO TO 43
        IF(ALS .GT.4.) GO TO 41
        LINEP(I) = 1H
        GO TO 41
43      CONTINUE
        TEM = RANF(IZL)
        VAL = ( ALS - 1.)/1.7777 +  .2*(TEM-0.5)
        IF(VAL .GT.0.1) GO TO 44
        LINEP(I) = 1H.
        GO TO 41
44      CONTINUE
        IF(VAL .GT..3) GO TO 45
        LINEP(I) = 1H+
        GO TO 41
45      CONTINUE
        IF(VAL .GT..5) GO TO 46
        LINEP(I) = 1HO
        GO TO 41
46      CONTINUE
        IF(VAL .GT..7) GO TO 47
        LINEP(I) = 1HX
        GO TO 41
47      CONTINUE
```

18

```
      LINEP(I) = 1H*
41    CONTINUE
      ITM = LNCV + 7
      IF(K .NE.ITM ) GO TO 51
      LINEP(65) = 1HG
      GO TO 40
51    CONTINUE
      ITM = LNCV + 8
      IF(K .NE.ITM ) GO TO 52
      LINEP(61) = 1HO
      LINEP(69) = 1HR
      GO TO 40
52    CONTINUE
      ITM = LNCV + 9
      IF(K .NE.ITM ) GO TO 53
      LINEP(57) = 1HR
      LINEP(73) = 1HA
      GO TO 40
53    CONTINUE
      ITM = LNCV + 10
      IF(K .NE.ITM ) GO TO 54
      LINEP(54) = 1HP
      LINEP(76) = 1HM
      GO TO 40
54    CONTINUE
      ITM = LNCV + 11
      IF(K .NE.ITM ) GO TO 55
      LINEP(23) = 1H
      LINEP(24) = 1H
      LINEP(106) = 1H
      LINEP(107) = 1H
      DO 63 JJ = 1,20
      JJJ = 4*JJ + 20
      DECODE(4,64,LABEL(JJ,1))KST(1),KST(2),KST(3),KST(4)
64    FORMAT(4A1)
      DO 65 JK = 1,4
      JJK = JJJ + JK
      LINEP(JJK) = KST(JK)
65    CONTINUE
63    CONTINUE
      GO TO 4[illegible]
55    CONTINUE
      ITM = LNCV + 12
      IF(K .NE.ITM ) GO TO 56
      LINEP(23) = 1H
      LINEP(24) = 1H
      LINEP(1[illegible]) = 1H
      DO 66 JJ = 1,20
      JJJ = 4*JJ + 20
      DECODE(4,64,LABEL(JJ,2))KST(1),KST(2),KST(3),KST(4)
      DO 67 JK = 1,4
      JJK = JJJ + JK
      LINEP(JJK) = KST(JK)
67    CONTINUE
66    CONTINUE
      GO TO 40
56    CONTINUE
      ITM = LNCV + 13
      IF(K .NE.ITM ) GO TO 57
      LINEP(56) = 1HR
      LINEP(74) = 1HO
      GO TO 40
57    CONTINUE
      ITM = LNCV + 14
      IF(K .NE.ITM ) GO TO 58
      LINEP(59) = 1HU
      LINEP(71) = 1HT
      GO TO 40
58    CONTINUE
      ITM = LNCV + 15
      IF(K .NE.ITM ) GO TO 59
      LINEP(63) = 1HB
      LINEP(67) = 1HA
      GO TO 40
59    CONTINUE
40    CONTINUE
      WRITE(MW,37)(LINEP(I),I= 1,129)
37    FORMAT(1H ,129A1)
21    CONTINUE


      WRITE(MW,77)
      IF(IFRSK .NE. 1) GO TO 121
      WRITE(MW,123)
      WRITE(MC,123)
123   FORMAT(///////)
      WRITE(MW,122)
122   FORMAT( ////1H0,30X, *  --- IN THIS RUN FALLOUT RISK CALCULATIONS
     1 WILL BE PERFORMED ---  *)
121   CONTINUE


      WRITE(MW,77)
77    FORMAT(1H1)

      ICTY1 = ICTYCB + 1
      WRITE(MW,140) ITELL(ICTY1)
140   FORMAT(         1H0,10X , A6,      * USE NNNN COUNTY CB AND CT RECORD
     1S FOR TARGET INPUT * )
      IBINT1 = IBINT + 1
      WRITE(MW,141) ITELL(IBINT1)
141   FORMAT(1H ,10X, A6,   * USE BINARY TAPE FOR TARGET INPUT *)
      IF(ILSTCT .GT. 0) GO TO  177
      WRITE(MW,179)
179   FORMAT(1H0,10X, *DO CALCULATIONS FOR ALL RECORDS ON INPUT TAPE*)
      GO TO 178
177   CONTINUE
      WRITE(MW,180) NSTIN,NCTIN
180   FORMAT( 1H0, 10X, *DO CALCULATIONS FOR ALL OF *, I2, * STATES AND
     1*, I3, * STATE COUNTY COMBINATIONS AS INPUT*)
178   CONTINUE
      ICLUS1=ICLUST+1
      WRITE(MW,110) ITELL(ICLUS1)
110   FORMAT(1H0,10X,A6,      * USE WEAPONS IN CLUSTERS AS INPUT*)
      ISIN1=ISINGW+1
      WRITE(MW,111) ITELL(ISIN1)
111   FORMAT(1H0,10X,A6,   * USE INDIVIDUAL WEAPONS AS INPUT DATA*)
      IF(ICLYLD .NE.0) GO TO 151
      WRITE(MW,152)
152   FORMAT(1H0, 10X,      * USE AVERAGED YIELDS AS INPUT FOR USE IN CL
     1USTER FALLOUT CALCULATIONS*)
      GO TO 160
```

```
151     CONTINUE
        IF(ICLYLD .NE. 1) GO TO 153
        WRITE(MQ,154)
154     FORMAT(1H0, 10X,      * USE FIRST YIELD INPUT AS ONLY YIELD FOR CL
       1USTER FALLOUT CALCULATIONS*)
        GO TO 160
153     CONTINUE
        IF(ICLYLD .NE.2) GO TO 155
        WRITE(MQ,156)
156     FORMAT(1H0, 10X,    * SELECT YIELD FROM CLOSEST OF SEVEN STANDARD C
       1LASSES FOR USE IN CLUSTER FALLOUT CALCULATIONS*)
        GO TO 160
155     CONTINUE
        WRITE(MQ,157)
157     FORMAT(1H0, 10X,     * PLACE INPUT YIELDS INTO CLASSES AS INPUT FOR
       1 USE IN CLUSTER FALLOUT CALCULATIONS*)
160     CONTINUE
        IF(MQCK .NE.1) GO TO 170
        IF(ICLYLQ .NE.0) GO TO 161
        WRITE(MQ,162)
162     FORMAT(1H0, 10X,       * USE AVERAGED YIELDS AS INPUT FOR USE IN QU
       1ICK WSEG FALLOUT CALCULATIONS *)
        GO TO 170
161     CONTINUE
        IF(ICLYLQ .NE. 1) GO TO 163
        WRITE(MQ,164)
164     FORMAT(1H0, 10X,        * USE FIRST YIELD INPUT AS ONLY YIELD FOR QU
       1ICK WSEG FALLOUT CALCULATIONS *)
        GO TO 170
163     CONTINUE
        IF(ICLYLQ .NE.2) GO TO 165
        WRITE(MQ,166)
166     FORMAT(1H0, 10X,    * SELECT YIELD FROM CLOSEST OF SEVEN STANDARD C
       1LASSES FOR USE QUICK WSEG FALLOUT CALCULATIONS*)
        GO TO 170
165     CONTINUE
        WRITE(MQ,167)
167     FORMAT(1H0, 10X,     * PLACE INPUT YIELDS INTO CLASSES AS INPUT FOR
       1 USE IN QUICK WSEG FALLOUT CALCULATIONS*)
170     CONTINUE


        WRITE(MQ,88)NSAMP
        IF(ISTRAT .EQ. 1) GO TO  85
88      FORMAT(1H0, *  THE SAMPLE SIZE = *, I5)
        WRITE(MQ,89)
89      FORMAT(1H0, * USE NON-STRATIFIED SAMPLES *)
        GO TO 86
85      CONTINUE
        WRITE(MQ,84)
84      FORMAT( 1H0, * USE SAMPLES STRATIFIED BY RADIUS AND ANGLE INTO 100
       1 EQUAL PROBABILITY AREAS* )
86      CONTINUE

        IF(IWNDST .EQ. 1) GO TO 120
        IF(IWNDST .EQ.2) GO TO 142
        WRITE(MQ,125) WIND,SGWP
125     FORMAT(1H0,   10X,       * FOR THIS RUN UNIFORM WIND STATISTICS WIL
       1L BE USED FOR THE ENTIRE COUNTRY*,/,15X, *AVERAGE WIND SPEED = *,
       2  F8.3, *, M. P. H.   STANDARD DEVIATION  OF WIND = * ,F8.3,
       3 * TIMES AVERAGE WIND SPEED*)
```

21

```
      WRITE(MQ,87) WMNDR
87    FORMAT (1H ,* THE MEAN WIND DIRECTION IS *, F10.4,   * DEGREES CLO
     1CKWISE FROM THE NORTH * )
      GO TO 127
126   CONTINUE
      WRITE(MQ,143) NUMO
143   FORMAT (1H0, 10X,        * USE WIND DATA FROM*, I3, * MONITOR POINTS
     1 FOR WIND STATISTICS *)

      WRITE(MQ,144) JSEA(ISEAS)
144   FORMAT(1H , 10X,  *USE * ,A6, * FOR SEASONAL WIND STATISTICS*)
      GO TO 127
142   CONTINUE
      WRITE(MQ,128)
128   FORMAT(1H0 ,10X,          * USE AN ALGORITHM TO CALCULATE WIND SPEED
     1 AS A FUNCTION OF POSITION FOR THE ENTIRE COUNTRY*)
127   CONTINUE
      IF( ISHRSW .NE.  0) GO TO 131
      WRITE(MQ,132) SHRST
132   FORMAT(1H ,10X,         * USE A CONSTANT SHEAR = *,F8.3, * M.P.H./KFT
     1.*)
      GO TO 133
131   CONTINUE
      IF(ISHRSW .NE. 1) GO TO 134
      WRITE(MQ,135) SHRST
135   FORMAT(1H  ,10X,         *USE SHEAR DISTRIBUTED AS A POSITIVE NORMAL
     1 DISTIRBUTION WITH MEAN SHEAR = * ,F8.3, * M.P.H./KFT.*)
      GO TO 133
134   CONTINUE
      WRITE(MQ,136) SHRST,SGSHRP
136   FORMAT(1H ,10X,       * USE NORMAL DISTRIBUTION FOR SHEAR WITH MEAN
     1 = *,  F8.4, * AND STANDARD DEV. = *, F8.4 , * TIMES THE MEAN*)
133   CONTINUE
      IF(IWLVSW .NE.0) GO TO 137
      WRITE(MQ,138)
138   FORMAT(1H ,10X, * FOR THIS CALCULATION USE ALL HORIZONTAL WINDS* )
      GO TO 139
137   CONTINUE
      IF(IWLVSW.NE.1) GO TO 147
      WRITE(MQ,148)
148   FORMAT(1H , 10X,     *USE WINDS LINEAR RELATIVE TO THE TARGET*)
      GO TO 139
147   CONTINUE
      IF(IWLVSW .NE.2) GO TO 149
      WRITE(MQ,150)
150   FORMAT(1H ,10X,     * USE RANDOM WIND CURVATURE RELATIVE TO THE TAR
     1GET  *)
      GO TO 139
149   CONTINUE
      WRITE(MQ,158)
158   FORMAT( 1H ,10X,  *  USE WIND COORDINATE SYSTEM FOR VARIATION WITH
     1 POSITION  *)
139   CONTINUE
      IF(MQCK .EQ. 1) GO TO 81
      WRITE(MQ ,83)
83    FORMAT (1H0, *  USE REGULAR WSEG 10 CALCULATION*)
      IF(MQCAL .NE. 0) GO TO  91
      WRITE(MQ'95)
95    FORMAT(1H0,  *USE WSEG BIOLOGICAL DOSE*)
      GO TO 94
91    CONTINUE
```

2

```
      IF(MUCAL .NE.1) GO TO 92
      WRITE(MQ,96)
96    FORMAT(1H0,  *USE NMCSSC BIOLOGICAL DOSE*)
      GO TO 94
92    CONTINUE
      IF(MUCAL .NE.2) GO TO 93
      WRITE(MQ,97)
97    FORMAT(1H0,  * USE NMCSSC MAXIMUM DOSE*)
      GO TO 94
93    CONTINUE
C     STOP666
94    CONTINUE
      GO TO 82
81    CONTINUE
      WRITE(MQ ,74)
74    FORMAT(1H0,  *   USE QUICK APPROXIMATION TO REGULAR WSEG 10 CALCUL
     1ATION*)
82    CONTINUE
      IF (ISEED .NE. 0) GO TO 73
      WRITE(MQ,75)
75    FORMAT(1H0, * USE CLOCK TO GIVE AUTOMATIC RANDOM NUMBER SEED*)
      GO TO 80
73    CONTINUE
      WRITE(MQ,70) SEED
70    FORMAT( 1H0,*USE RANDOM NUMBER SEED OF*, F16.6)
80    CONTINUE
      IF(MSDSW  .NE.1)GO TO 145
      WRITE(MQ,146) CSEED
146   FORMAT(1H0, 10X,      * USE THE SAME STRING OF RANDOM NUMBERS FOR E
     1ACH TARGET INITIATED BY A SEED OF*,F15.4)
145   CONTINUE
      IF(IFRCAS .EQ.1) GO TO 171
      WRITE(MQ,172)
172   FORMAT(1H0,  10X, * DO NOT DETERMINE FALLOUT FATALITIES IN RISK CA
     1LCULATIONS *)
      GO TO 173
171   CONTINUE
      WRITE(MQ,174)
174   FORMAT( 1H0, 10X *DO FATALITY AND CASUALTY CALCULATIONS IN RISK CA
     1LCULATIONS AT THE FOLLOWING PROTECTION FACTOR LEVELS *./,
     2 1H0, 20X,14HURBANIZED AREA  , 23X, 10HRURAL AREA  ,/,
     3  1H ,5X, 4HNO.      ,5X, 10HPROT. FAC.   ,5X, 10H FRACTION    ,
     4 10X, 10HPROT. FAC.     ,5X, 10H FRACTION      )
      DO 176 I = 1,NPFLV
      WRITE(MQ,175) I, PFLVU(I), FRPFU(I),PFLVR(I),FRPFR(I)
175   FORMAT(1H , 5X,1H(,I2,1H), 5X, F10.2,5X,F10.4,10X,F10.2,5X,F10.4)
176   CONTINUE
173   CONTINUE
      WRITE(MQ,77)
      RETURN
      END


      SUBROUTINE CLSIN


      COMMON/RUNSW/ICTYCB,IBINT,ICLUST,ISTNSW,LABEL(20,2),ISPWSW,ILGWSW,
     1IEVWSW,IMOST,NUSEV,IMSWSW,ILVSW,DOSELV(5),IPROSW,PROLV(3),ICATSW,
     2 MOSTP,MLVTP,MMPTP,TCLSA,JREST,JRESTN,JRESTA,JRESTD,JRESTE,JRESTF,
     3 IFRSK,IUMAS,NSAMP,ISTRAT,IWNDST,ISHRSW,IWLVSW,MUCK,ISEED,SEED,
     4 MSDSW,CSEED,IFRCAS,NPFLV,PFLVU(7),PFLVR(7),FDSEM,FDSESG,CDSEM,
5 CDSESG,MDCAL,ICLYLD,ICLYLG,FRPFU(7) ,FRPFR(7)
6,ILSTCT, ISTIN(50),ICTIN(100),ISEAS,NSTIN,NCTIN                    23
 COMMON/CLSPR/ YLDC(500),FISSC(500),XWC(500),YWC(500),SIGXC(500),
1 SIGYC(500),NCLS,SARY(500,5),YLDI(7),YLDM(6),SARYL(7,19),
2ARRY(7,40)
 COMMON/CLFLPR/YIELD,FISS,WINN,SHN,XL,SIGW,DWN,CRS,DOSE,STR(5)
 COMMON/CARTOG/IC,JC,KC,FLATP,FLONP,Y,X,RHO,THETA, FLATC,FLONC,
1SCALE, STCHOR,STCHAR,HILATR,ROLATR,XGOSTR,YGOSTR,XMPOFR,YMPOFR,
2 CRMPLA,CRMPLO
 COMMON/IOPR/ MP,MQ,MS,NA,NC,ND,NE,NF,NG,NH,NI,NJ,NK,NL,NM
 COMMON/LAZY/TTA,TTB,TTC,TTD,TTE,KTA,KTB,KTC,KTD,KTE,KTF,KTG,KTH
 COMMON/LAZYA/TTAA(200),KTAA(200)
 DATA(TOSMI= 69.171339),(TORAD = 0.01745329)

C        INITIALIZE FOR RECTANGLE PROJECTION
         IC = 3
         KC = I
         JC = 2
         CALL PROJCT
         JC = 1
         NCLCT = 1

         I = 0
         SUMYF = 0.
         SUMYLA =0.
         SUMYLO = 0.
         SUMNWP = 0.
10       CONTINUE
         I = I + 1
         READ (NC,11)YLDC(I),            XWLON,YWLAT,SIGXS, SIGYS,
       ' XMNLN,XMXLN,XMNLT,XMXLT,FISSC(I),XWP
11       FORMAT(10F7.3,1X,F4.0)
         IF(YLDC(I) .LT. .) GO TO 15
         FLATP = YWLAT
         FLONP = XWLON
         CALL PROJCT            To set to a rectangular coordinate system
         XWC(I) = X
         YWC(I) = Y
         SIGXC(I) = SIGXS*TOSMI*COS(TORAD*YWLAT)
         SIGYC(I) = SIGYS*TOSMI
         TEMP = YLDC(I)*FISSC(I)
         SUMYF = SUMYF + TEMP
         SUMYLA = SUMYLA + Y*TEMP
         SUMYLO = SUMYLO + X*TEMP
         SUMNWP = SUMNWP + XWPN

C        CALCULATE YIELD DEPENDENT PARAMETERS FOR CLUSEER WSEG 1.
         IF(ICLYLD .NE.0) GO TO 21
         YIELD = YLDC(I)
         FISS = FISSC(I)
         CALL CFALLY
         DO 12 J = 1,5
         SARY(I,J) = STR(J)
12       CONTINUE

         GO TO 10
21       CONTINUE
         IF(ICLYLD .NE.1) GO TO 22
         IF(I.NE.1) GO TO 10
```

```
      YIELD = YLDC(I)
      FISS = FISSC(I)
      CALL CFALLY
      DO 23 J = 1,5
      SARY(1,J) = STR(J)
23    CONTINUE
      GO TO 10
22    CONTINUE
      IF(ICLYLD .NE.3) GO TO 10
      DO 24 K = 1,NCLCT
      IF(YLDC(I) .EQ. YLDI(K) ) GO TO 10
24    CONTINUE
      YLDI(NCLCT) = YLDC(I)
      YIELD = YLDC(I)
      FISS = FISSC(I)
      CALL CFALLY
      DO 27 L = 1,5
      SARY(NCLCT,L) = STR(L)
27    CONTINUE
      NCLCT = NCLCT + 1
      IF (NCLCT .LE.7) GO TO 10
      STOP 4563
15    CONTINUE
      NCLS = I - 1
      ITM = SUMNWP
      TLA = SUMYLA/SUMYF
      TLO = SUMYLO/SUMYF
      WRITE(MW,31) ITM, NCLS    ,SUMYF, TLA,TLO
31    FORMAT(///, 10X,*THE WEAPON CLUSTER LIST HAS *,I5, * WEAPONS IN *
     1 ,I4, * CLUSTERS WITH A TOTAL FISSION YIELD = *, F10.2, * MT.*
     2 ,/,15X, *THE WEIGHTED C.G. HAS NORTHING = *,F10.4, * AND EASTING
     3= * ,F10.4)
      RETURN
      END

      SUBROUTINE TGTIN

C     READS TARGET LOCATIONS

      DIMENSION IZL(15)
      COMMON/RUNSW/ICTYCB,IBINT,ICLUST,ISINGW,LABEL(20,2),ISPWSW,ILGWSW,
     1IEVWSW,DMOST,NOSEV,IHSWSW,ILVSW,DOSELV(5),IPROSW,PROLV(3),ICATSW,
     2 MOSTP,MLVTP,MMFTP,ICLSQ,JREST,JRESTN,JRESTA,JRESTD,JRESTE,JRESTF,
     3 IFRSK,IUMAS,NSAMP,ISTRAT,IWNDST,ISHRSW,IWLVSW,MWCK,ISEED,SEED,
     4 MSDSW,CSEED,IFRCAS,NPFLV,PFLVU(7),PFLVR(7),FDSEM,FDSESG,CDSEM,
     5 CDSESG,MDCAL,ICLYLD,ICLYLG,FRPFU(7) ,FRPFR(7)
     6,ILSTCT, ISTIN(50),ICTIN(100),ISEAS,NSTIN,NCTIN
      COMMON/RUNPR/ XNSAMP,NRPIS,NSMPCT,ILAST,ISTNO,ISTLO,JSTCD,JSTNM,
     1 ISTC(51),ISTNM(51),ICARRY(100),DSEPLT,IDSEPT,XPLT,YPLT
     2,DSEPLA,DSEPLB,DSEPLC,ISWING,JSWING,FRFATU,FRCASU,FRFATR,FRCASR
      COMMON/TARPR/ NAMETR(10),TRPT,TRPU,TRPR,TRPUA,IRSTCO,IRCUCO,
     1IRUAC,IREAC,ISMSAC,NUTR,TRCGLA,TRCGLO,YTL,XTL,SIGTRB,SIGTRL,
     2ALTR,NPLCS,NMCUS,NTT(52),PUTT(52),PTIT(52),PRTT(52),PATT(52)
      COMMON/WNDPR/WIND,V,SGWP,SGW,SHRST,SHR,SGSHRP,SGSHR,WMNDR,ALWND,
     1 CAL,SAL, FLATMO(100),FLONMO(100),DEGMO(100),SPDMO(100),VSMO(100),
     2 NUMO
      COMMON/CARTOG/IC,JC,KC,FLATP,FLONP,Y,X,RHO,THETA, FLATC,FLONC,
     1SCALE, STCHUR,STCHAR,HILATR,ROLATR,XGOSTR,YGOSTR,XMPOFR,YMPOFR,
```

Group yield
into ≤ 7 classes

25

```
     2 CRMPLA,CRMPLO
       COMMON/TOPR/ MP,MQ,MS,NA,NC,ND,NE,NF,NG,NH,NI,NJ,NK,NL,NM
       COMMON/LAZY/ITA,ITB,ITC,ITD,ITE,KIA,KIB,KIC,KID,KTE,KTF,KIG,KTH
       COMMON/LAZYA/ITAA(200),KTAA(200)


       ILAST = 0
       IF(ICTYCB .NE. 1 .AND.ICTYCB .NE. 2) GO TO 50
       IF(ILSTCT .NE. 0)  GO TO 40


C      INPUT FROM NNNN
C      NATIONWIDE DATA WITH EOF,S ON IDA TAPE1818
C      NATIONWIDE DATA WITHOUT EOF,S AFTER EACH STATE ON IDA TAPE 1853
C      TEXAS ONLY DATA ON IDA TAPE NO. 1189
10     CONTINUE
       READ(NA,12) LETCB,IRSTCO, IRCOCO, NAMETR(1),NAMETR(2),NAMETR(3),
     1 NAMETR(4), TRCGLO,TRCGLA, SIGTRB,SIGTRL,ALTR, TRPT,ANSEG,LNAT
12     FORMAT( A4,A2,A3,4X,3A8,A6, 5F8.4,F10.0, F8.4,30X,A4)
C **********3600/6400 COMPATABILITY CHANGE **************
       IFL = EOF(NA)
       IF(IFL .EQ.0) GO TO 15
C      IF(EOF,NA) 16,15
16     CONTINUE
       IF(ICTYCB .EQ.2) GO TO 17
       ILAST = 1
       RETURN
17     CONTINUE
       IF (IRSTCO .NE.2H56) GO TO 18
       ILAST = 1
       RETURN
18     CONTINUE
       IF(LNXT .EQ.2) GO TO 15
       IF(ILSTCT .NE.0) GO TO 44
       GO TO 10
15     CONTINUE
       READ(NA,13) LTRCT, TRPU,TRPR,TRPUA, NUREDS, NRUEDS, NUAEDS,
     1 HSEUR, HSERU, HSEUA, NPLCS, NMCUS,ISMSAC,IREAC,IRUAC,LTNAT
13     FORMAT(  A4,9X,3F10.0,3I6, 3F8.0,I4,I4,A4,A5,4X,A4,21X,A4)
C **********3600/6400 COMPATABILITY CHANGE **************
       IFL = EOF(NA)
       IF(IFL .NE.0) GO TO 19
       GO TO 20
C      IF(EOF,NA) 19,20
19     CONTINUE
       LNAT = 2
       GO TO 16
C
40     CONTINUE
       READ(NA,41) LETCB,IRSTCO,IRCOCO,(IZL(J),J = 1,9)
C***************3600/6400 COMPATIBILITY***************
 41    FORMAT( A4,A2,A3,4X,8A10,A8)
C41    FORMAT( A4,A2,A3,4X,11A8)
       IFL = EOF(NA)
       IF(IFL .NE. 0 ) GO TO 16
C      IF(EOF,NA) 16,46
46     CONTINUE
C

       IF (NSTIN .EQ. 0 ) GO TO 33
       DO 32 JJ = 1,NSTIN
```

Bad Target Data

```
      IF(IRSTCO .EQ. ISTIN(JJ)) GO TO 31
32    CONTINUE
33    CONTINUE
      IF(ILSTCT .EQ.1 .OR. NCTIN .EQ.0) GO TO 44
      ENCODE(5,36,ITEMP ) IRSTCO,IRCOCO
36    FORMAT(A5,A3)
      DO 34 JJ = 1,NCTIN
      IF(ITEMP .EQ. ICTIN(JJ) ) GO TO 31
34    CONTINUE
44    CONTINUE
      NUTR = NUTR + 1
      READ(NA,42)
42    FORMAT(1X)
      GO TO 40
31    CONTINUE
      DECODE(88,43,IZL)              NAMETR(1),NAMETR(2),NAMETR(3),
     1 NAMETR(4), TRCGLO,TRCGLA, SIGTRB,SIGTRL,ALTR, TRPT,ANSEG,LNXT
43    FORMAT(3A8,A6,5F8.4,F10.0,F8.4)
      READ(NA,13) LTRC1, TRPU,TRPR,TRPUA, NUREDS, NRUEDS, NUAEDS,
     1 HSEUR, HSERU, HSEUA, NPLCS, NMCDS,ISASAC,IREAC,IRUAC,LTNXT
```

Read Target Data

```
20    CONTINUE
      IC = 3
      JC = 2
      KC = 1
      CALL PROJCT
```

Put target in rectangular coordinate system

```
      JC = 1
      FLATP = TRCGLA
      FLONP = TRCGLO
      CALL PROJCT
      XTL = X
      YTL = Y
```

target coordinates

```
      IF(IWNDST .NE.1) GO TO 21
      COSLA=COS(TRCGLA*3.14159265/180.0)
      XMIN=99999999.
      DO 22 J=1,NUMO
      DELA=TRCGLA-FLATMO(J)
      DELOM=(TRCGLO-FLONMO(J))*COSLA
      DSQ=DELA*DELA+DELOM*DELOM
      IF(DSQ.GT.XMIN) GO TO 22
      JJ=J
      XMIN = DSQ
  22   CONTINUE
      WIND=SPDMO(JJ)
      WMNDR=DEGMO(JJ)
      SGW = VSMO(JJ)
```

find wind statistics for target

```
  21  CONTINUE
50    CONTINUE
      RETURN
      END

      SUBROUTINE BOXFL

C     DOES THE CHORE OF ACCUMULATING THE VARIOUS STATISTICS

C*****TEMPORARY*****
      DIMENSION FTIND(100),CSIND(100)
      COMMON/RUNSW/ICTYCB,IBIAT,ICLUST,ISINW,W,LABEL(20,2),ISPWSW,ILGWSW,
```

wind statistics

27

```
     1IEV*SW,DMOST,NOSEV,IHSWSW,ILVSW,DOSELV(5),IPRNSW,PROLV(3),ICATSW,
     2 MDSTP,MLVTP,MMPTP,ICLSA,JREST,JRESTN,JRESTA,JRESTD,JRESTE,JRESTF,
     3 IFHSK,IDMAS,NSAMP,ISTRAT,IWNOST,ISHRSW,IWLVSW,MUCK,ISEED,SEED,
     4 MSDSW,CSEED,IFHCAS,NPFLV,PFLVD(7),PFLVR(7),FDSEM,FDSESG,CDSEM,
     5 CDSESG,MDCAL,ICLYLD,ICLYLG,FRPFD(7) ,FRPFR(7)
     5,ILSICT, ISTIN(50),ICTIN(100),ISEAS,NSTIN,NSTIN
      COMMON/RUNPR/ XNSAMP,NRPIS,NSMPCT,ILAST,ISTNO,ISTLO,JSTCD,JSTNM,
     1 ISTC(51),ISTNM(51),ICARRY(100),DSEPLT,IDSEPT,XPLT,YPLT
     2,DSEPLA,DSEPLB,DSEPLC,ISWING,JSWING,FRFATU,FRCASU,FRFATR,FRCASR
      COMMON/TARPR/ NAMETR(10),TRPT,TRPU,TRPR,TRPUA,IRSTCO,IRCOCO,
     1IRUAC,IREAC,ISMSAC,NUTR,IRCGLA,TRCGLO,YTL,XTL,SIGTRB,SIGTRL,
     2ALTR,NPLCS,NMCDS,NTT(52),PUTT(52),PTTT(52),PRTT(52),PATT(52)
      COMMON/STAPR/DTOT,DMAX,BV(51),BE(502),XLBV(51),NB(51),NBE(501),
     1NBL(50),NBOXA,NBOXL,NBOXE,SMLDOS,SD,SDS,SDC,SDF,SSD,SSDS,SSDC,SSDF
     2,NSML,NBAST(50,52),FBXST(100,52),CHXST(100,52),PROFT(50),PROCS(50)
     3,SFT,SFTS,SFTC,SFTF, SCA, SCAS,SCAC,SCAF,TPOPLV(50,52)
     4,DSEIND(100)
      COMMON/IOPR/ MP,MG,MS,NA,NC,ND,NE,NF,NG,NH,NI,NJ,NK,NL,NM
      COMMON/LAZY/ITA,ITB,ITC,ITD,ITE,KIA,KIB,KIC,KTD,KIE,KTF,KTG,KTH
      COMMON/LAZYA/TTAA(200),KTAA(200)
C*****TEMPORARY*****
      EQUIVALENCE(FTIND(1), TTAA(1)),(CSIND(1),TTAA(101))
```

Test Program LASH

```
      SD = SD + DTOT
      TMA = DTOT*DTOT
      SDS = SDS + TMA
      TMB = TMA*DTOT
      SDC = SDC + TMB
      TMC = TMB*DTOT
      SDF = SDF + TMC
      IF(DTOT .LT. SMLDOS) GO TO 72
      SSD = SSD + DTOT
      SSDS = SSDS + TMA
      SSDC = SSDC + TMB
      SSDF = SSDF + TMC
      GO TO 73
72    CONTINUE
      NSML = NSML + 1
73    CONTINUE
      IF(DMAX .LT. DTOT) DMAX = DTOT
      DO 61 J = 1,NBOXL
      IF(   DTOT .GE. XLBV(J) ) GO TO 61
      NBL(J - 1) = NBL(J-1) + 1
      GO TO 62
61    CONTINUE
      NBL(NBOXL) = NBL(NBOXL) + 1
62    CONTINUE
      DO 63 J = 1, NBOXA
      IF( DTOT .GE. BV(J) ) GO TO 63
      NB(J-1) = NB(J-1) + 1
      NBXST(J-1,ISTNO) = NBXST(J-1,ISTNO)+ 1
      NBXST(J-1,52) = NBXST(J-1,52) + 1
      TPOPLV(J-1,ISTNO) = TPOPLV(J-1,ISTNO) + TRPT
      TPOPLV(J-1'52) = TPOPLV(J-1'52) + TRPT
      GO TO 64
63    CONTINUF
      NB(NBOXA) = NB(NBOXA) + 1
      NBAST(NBOXA,ISTNO) = NBAST(NBOXA,ISTNO) + 1
      NBAST(NBOXA,52) = NBAST(NBOXA,52) + 1
      TPOPLV(NBOXA,ISTNO) = TPOPLV(NBOXA,ISTNO) + TRPT
```

28

```
      TPOPLV(NBOXA,52) = TPOPLV(NBOXA,52) +  TRPT
64    CONTINUE
      DO 65 J = 1,NBOXE
      IF(DTOT .GE. BE(J) )  GO TO 65
      NBE(J-1) = NBE(J-1) + 1
      GO TO 66
65    CONTINUE
      NBE(NBOXE) = NBE(NBOXE) + 1
66    CONTINUE
      IF(NSMPCT .GT.100) GO TO 51
      DSEIND(NSMPCT) = DTOT

      IF(IFRCAS .NE.1) GO TO 51
      FRFATU = 0.
      FRFATR = 0.
      FRCASU = 0.
      FRCASR = 0.
      DO 52 J = 1,NPFLV
      DSEFU = DTOT/PFLVU(J)
      DSEFR = DTOT/PFLVR(J)
      Z =(DSEFU -FDSEM)/FDSESG
      FRFATU = FRFATU + CUMNOR(Z)*FRPFU(J)
      Z =(DSEFR -FDSEM)/FDSESG
      FRFATR = FRFATR+ CUMNOR(Z)*FRPFR(J)
      Z = (DSEFU - CDSEM)/CDSESG
      FRCASU = FRCASU + CUMNOR(Z)*FRPFU(J)
      Z = (DSEFR - CDSEM)/CDSESG
      FRCASR = FRCASR + CUMNOR(Z)*FRPFR(J)
52    CONTINUE
      TEMPA = (TRPT- TRPUA)*FRFATR + TRPUA*FRFATU
      TEMPB = (TRPT - TRPUA)*FRCASR + TRPUA*FRCASU
      FAXST(NSMPCT,ISTNO) = FAXST(NSMPCT,ISTNO)+TEMPA
      FAXST(NSMPCT, 52  ) = FAXST(NSMPCT, 52  )+TEMPA
      CAXST(NSMPCT,ISTNO) = CAXST(NSMPCT,ISTNO) +TEMPB
      CAXST(NSMPCT, 52  ) = CAXST(NSMPCT, 52  ) +TEMPB
C*****TEMPORARY*****
      ILSFSW = KTA
      IF( ILSFSW .NE.1) GO TO 21
      FTIND( NSMPCT) = TEMPA
      CSIND(NSMPCT) = TEMPB
21    CONTINUE
51    CONTINUE
      RETURN
      END
```

Accumulate Fallout Fatality Statistics

```
      SUBROUTINE WRTONE

C     PERFORMS THE FINAL OUTPUT OF DOSE STATISTICS FOR EACH TARGET

      DIMENSION LOHDST(100)
C*****TEMPORARY*****
      DIMENSION FTIND(100),CSIND(100)

      DIMENSION TDBX(10)
      COMMON/RUNSW/ICTYCB,IBINT,ICLUST,ISINGW,LABEL(20,2),ISPASW,ILGWSW,
     1IEVWSW,DMOST,NDSEV,IHSWSW,ILVSW,DOSELV(5),IPROSW,PROLV(5),ICATSW,
     2 NDSTP,MLVTP,MMPTP,ICLSW,JREST,JRESTN,JRESTA,JRESTD,JRESTE,JRESTF,
     3 IFRSK,IOMAS,NSAMP,ISTRAT,IWNDST,ISHRSW,IWLVSW,MWCK,ISEED,SEED,
```

Similar to Subroutine WRTLST of Program LASM

29

```
      * WSDSW,CSEED,IFWCAS,NPFLV,PFLVU(7),PFLVR(7),FDSEM,FDSESG,CDSEM,
      5 CDSESG,MDCAL,ICLYLD,ICLYLG,FRPFU(7) ,FRPFR(7)
      5,ILSTCT, ISTIN(50),ICTIN(100),ISEAS,NSTIN,NCTIN
       COMMON/RUNPR/ XNSAMP,NRPTS,NSMPCT,ILAST,ISTNO,ISTLO,JSTCD,JSTNM,
      1 ISTC(51),ISTNM(51),ICARRY(100),DSEPLT,IDSEPT,XPLT,YPLT
      2,DSEPLA,DSEPLB,DSEPLC,ISWING,JSWING,FRFATU,FRCASU,FRFATR,FRCASR
       COMMON/TARPR/ NAMETR(10),TRPT,TRPU,TRPR,TRPUA,IRSTCO,IRCOCO,
      1IRUAC,IREAC,ISMSAC,NUTR,TRCGLA,TRCGLO,YTL,XTL,SIGTRB,SIGTRL,
      2ALTR,NPLCS,NMCDS,NTT(52),PUTT(52),PTTT(52),PRTT(52),PATT(52)
       COMMON/WNDPR/WIND,V,SGWP,SGW,SHRST,SHR,SGSHRP,SGSHR,WMNDR,ALWND,
      1 CAL,SAL, FLATMO(100),FLONMO(100),DEGMO(100),SPDMO(100),VSMO(100),
      2 NUMO
       COMMON/STAPR/DTOT,DMAX,BV(51),BE(502),XLBV(51),NB(51),NBE(501),
      1NBL(50),NBOXA,NBOXL,NBOXE,SMLDOS,SD,SZS,SDC,SDF,SSD,SSDS,SSDC,SSDF
      2,NSML,NBAST(50,52),FBAST(100,52),CBXST(100,52),PROFT(50),PROCS(50)
      3,SFT,SFIS,SFTC,SFTF, SCA, SCAS,SCAC,SCAF,TPOPLV(50,52)
      4,DSEIND(100)
       COMMON/CARTOG/IC,JC,KC,FLATP,FLONP,Y,X,RHO,THETA, FLATC,FLONC,
      1SCALE, STCHUR,STCHAR,HILATR,ROLATR,XGDSTR,YGDSTR,XMPOFR,YMPOFR,
      2 CRMPLA,CRMPLO
       COMMON/IOPR/ MP,MQ,MS,NA,NC,ND,NE,NF,NG,NH,NI,NJ,NK,NL,NM
       COMMON/LAZY/TTA,TTB,TTC,TTD,TTE,KTA,KTB,KTC,KTD,KTE,KTF,KTG,KTH
       COMMON/LAZYA/TTAA(200),KTAA(200)
C*****TEMPORARY*****
       EQUIVALENCE(FTIND(1), TTAA(1)),(CSIND(1),TTAA(101))

       NSSUM = 0
       NLSUM = 0
       NFSUM = 0
77     FORMAT(1H1)

       IF(ICATSW .NE.1) GO TO 39
       IF(ISWING .EQ.2) GO TO 37
       IF(ILGWSW .EQ.1 .OR.IEVWSW .EQ.1) GO TO 37
       ISWING = 2
       WRITE(MQ,74)
74     FORMAT(1H0)
       GO TO 38
37     CONTINUE
       WRITE(MQ,77)
       ISWING = 1
38     CONTINUE

       WRITE(MQ,75)  (NAMETR(I) ,I = 1,4),   JSTNM     ,IRSTCO,IRCOCO,
      1TRPT,TRPU,TRPR,TRCGLA,TRCGLO,YTL,XTL,NUTR
75     FORMAT(1H0, *  FALLOUT DOSE DISTRIBUTIONS FOR *,
      1 3A8,A6,  * IN THE STATE *,A2,* WITH CODE *,A2,* WITH CTY CODE *,
      2 A3,/,* TARGET POPULATION--- TOTAL = *, F10.0, *   URBAN = *,
      3F10.0,  *    RURAL = *, F10.0,
      4/,*  LATITUDE = *, F9.4,  *    LONGITUDE = * ,F9.4, * NORTHING = *,
      5F9.2,*  EASTING = *,F9.2, *    TARGET NO. = *,I4)
       IF(TRPUA. EQ. 0.) GO TO 69
       WRITE(MQ,68) IRUAC,TRPUA
68     FORMAT( 1H ,*  THIS COUNTY HAS AN URBANIZED AREA WITH CODE *, A4,
      1 * AND URBANIZED AREA POPULATION = *, F10.0)
69     CONTINUE
       IF(IWNDST. EQ.0) GO TO 53
       WRITE(MQ,54)WMNDR,WIND,SGW
54     FORMAT(1H , * FOR THIS TARGET MEAN WIND DIRECTION = *, F7.3,
      1 * DEGREES, SPEED = *, F7.3,* MPH.   VECTOR STD. DEV. = *, F7.3,
      2  * MPH.*)
```

Header

Unit Dose Statistics

```
53      CONTINUE
39      CONTINUE
        WRITE(MQ,78) DMAX
78      FORMAT( 7H0, * THE MAXIMUM DOSE =* F13.4   )
        XMN = SD/XNSAMP
        XMNS = XMN*XMN
        XNUS = SDS/XNSAMP
        VAR = XNUS - XMNS
        STD = SQRT(ABS(VAR))
        IF(STD .GT. 0.00001) GO TC 31
        XNUC = 0.
        XKUR = 0.
        GO TO 32
31      CONTINUE
        XNUC = SDC/XNSAMP
        SKW = (XNUC - 3.*XNUS*XMN + 2.*XMNS*XMN)/(2.*VAR*STD)
        XNUF = SDF/XNSAMP
        XKUR=( XNUF - 4.*XNUC*XMN + 6.*XNUS*XMNS - 3.*XMNS*XMNS)/(VAR*VAR)
32      CONTINUE
        WRITE(MQ,76)XMN,STD,SKW,XKUR
76      FORMAT(1H ,*DOSE STATISTICS -- MEAN = *, F10.4,* STANDARD DEVIATIO
       1N = *, F10.4,  * SKEWNESS =*,F10.4,* KURTOSIS =*,F10.4    )
        IF(NSML.NE.NSAMP) GO TO 48
        WRITE(MQ,49) SMLDOS
49      FORMAT( 1H ,* ALL DOSES LESS THAN *,F10.1)
        GO TO 46
48      CONTINUE
        IF(NSML .GT.0) GO TO 41
        WRITE(MQ,42)SMLDOS
42      FORMAT( 1H ,  * NO DOSES LESS THAN *, F10.1 )
        GO TO 46
41      CONTINUE
        IF(NSML .GT. 1) GO TO 43
        TEMP  = SD - SSD
        WRITE(MQ,44) TEMP,SMLDOS
44      FORMAT( 1H , * ONE DOSE OF*, F10.2 , * IS LESS THAN DOSE OF*,
       1  F10.2)
        GO TO 46
43      CONTINUE
        XNSML = NSML
        XNSML = XNSAMP - XNSML
        XMN = SSD/XNSML
        XMNS = XMN*XMN
        XNUS = SSDS/XNSML
        VAR = XNUS - XMNS
        STD = SQRT(VAR)
        IF(STD .GT. 0.00001)  GO TO 33
        XNUC = 0.
        XKUR = 0.
        GO TO 34
33      CONTINUE
        XNUC = SSDC/XNSML
        SKW = (XNUC - 3.*XNUS*XMN + 2.*XMNS*XMN)/(2.*VAR*STD)
        XNUF = SSDF/XNSML
        XKUR=( XNUF - 4.*XNUC*XMN + 6.*XNUS*XMNS - 3.*XMNS*XMNS)/(VAR*VAR)
34      CONTINUE
        WRITE(MQ,47) NSML    ,SMLDOS, XMN,STD,SKW,XKUR
47      FORMAT( 1H , * DOSE STATISTICS EXCLUDING * ,I5  , * DOSES UNDER *,
       1 F8.0, * R. ---  *,/, 10X, *MEAN = *,F10.4, * STANDARD DEVIATION =
       2 *, F10.4,  * SKEWNESS = *, F10.4, * KURTOSIS = * ,F10.4)
```

3)

2nd Percentile Levels

```
46      CONTINUE
C
C       THIS ASSUMES ONLY 100 DOSES IN SAMPLE**************
        IF(NSAMP .GT. 100) GO TO 61
        IF(IPROSW .NE. 1) GO TO 51
        MIDNU = NSAMP /2
        LOWNU = NSAMP/10
        NHINU = NSAMP - LOWNU
        XLOW = 0.1
        XMID = 0.5
        XHIGH = 0.9
        GO TO 52
51      CONTINUE
        IF(IPROSW.NE.2) GO TO 61
        TEMP = PROLV(1)*XNSAMP
        LOWNU = TEMP
        TEMP = PROLV(2)*XNSAMP
        MIDNU = TEMP
        TEMP = PROLV(3)*XNSAMP
        NHINU = TEMP
        XLOW = PROLV(1)
        XMID = PROLV(2)
        XHIGH = PROLV(3)
52      CONTINUE
        CALL FORD(DSEIND,LORDST,100)
        JTEM = LOWNU + 1
        ITM = LORDST(JTEM)
        DSEL = DSEIND(ITM)
        JTEM = LOWNU
        ITM = LORDST(JTEM)
        TTM = DSEIND(ITM)
        DSEL = 0.5*(DSEL + TTM)
        JTEM = MIDNU + 1
        ITM = LORDST(JTEM)
        DSEM = DSEIND(ITM)
        JTEM = MIDNU
        ITM = LORDST(JTEM)
        TTM = DSEIND(ITM)
        DSEM = 0.5*(DSEM + TTM)
        JTEM = NHINU + 1
        ITM = LORDST(JTEM)
        DSEH = DSEIND(ITM)
        JTEM = NHINU
        ITM = LORDST(JTEM)
        TTM = DSEIND(ITM)
        DSEH = 0.5*(DSEH + TTM)
        WRITE(MG,66) ALOW,DSEL,XMID,DSEM,XHIGH,DSEH
66      FORMAT(1H0, *THERE IS A *, F4.2, * PROBABILITY OF A DOSE LESS THAN
       1 * ,F10.0,/, *     A * ,F4.2, * PROBABILITY OF A DOSE LESS THAN *,
       2 F10.0,/, *     A *,F4.2, * PROBABILITY OF A DOSE LESS THAN *,
       3 F10.0 )
        DSEPLT = DSEM
        DSEPLA = DSEL
        DSEPLB = DSEM
        DSEPLC = DSEH
61      CONTINUE
C
        IF(MMPTP .NE.1) GO TO 62
C       ASSUMES NO MORE THAN 100 SAMPLES
        IF(NSAMP .GT. 100) GO TO 62
```

Detailed output of each dose

3✓

```
      WRITE(NF,63) IRSTCO,IRCOCC,NUTR,TRCGLA,TRCGLO,TRPT,TRPU,TRPUA
63    FORMAT(A2,1X,A3,I6,2F10.5,3F10.0)
      ITM = NSAMP/10
      ITTM = ITM*10
      IF(ITM .NE.ITTM) ITM = ITM + 1
      JJA = -10
      DO 64 JJ = 1,ITM
      JJA = JJA + 10
      DO 65 JJB = 1,10
      JJC = JJA + JJB
      JTM = LORDST(JJC)
      TDBX(JJB) = USEIND(JTM)
65    CONTINUE
      WRITE(NF,67) (TDBX(J),J = 1,10)
67    FORMAT(10F11.3)
64    CONTINUE
62    CONTINUE


      IF(ISPWSW .NE.1) GO TO 79
      WRITE(MG,71)
71    FORMAT( 1H0, *INTERVAL VALUES *//,10X, 6H NO.    10X, 10HMIN. VALU
     1E    ,10X,  10HMAX. VALUE    , 10X,   10H NUMBER    ,10X,
     2  10H CUML. NO.     )
      DO 72 J = 1,NBOXA
      JJ = J + 1
      NSSUM = NSSUM + NB(J)
      WRITE(MG,73)  J , BV(J) , BV(JJ), NB(J )      ,NSSUM
73    FORMAT( 1H  , 10X, 1H( , I2, 1H), 10X, F10.0,10X, F10.0, 10X,
     1 I10  ,10X,I10 )
72    CONTINUE
79    CONTINUE
      IF(MUSTP .NE.1) GO TO 35
      WRITE(NO,36) IRSTCO,IRCOCC,NUTR, TRCGLA,TRCGLO,TRPT,
     1 (NB(J), J = 1,NBOXA)
36    FORMAT(A2,1X,A3,I6,2F10.5,F10.0,9I6)
  35  CONTINUE
      IF(ILGWSW .NE. 1) GO TO 82
      WRITE(MG,81)
81    FORMAT( 1H1, * EQUAL LOGARITHMIC INTERVALS *,////,
     1 10X, 6H  NO.    , 10X, 10HMIN. VALUE   , 12X, 10HMAX. VALUE,
     2  12X,  10H NUMBER     ,10X, 10H CUML. NO.       )
      DO 83 J = 1,NBOXL
      JJ = J + 1
      NLSUM = NLSUM + NBL(J)
      WRITE(MG,84) J, XLBV(J) , XLBV(JJ), NBL(J     ) ,NLSUM
84    FORMAT( 1H  , 10X, 1H(, I2, 1H), 10X, F12.2, 10X, F12.2,10X,I10,
     1    10X, I10)
83    CONTINUE
82    CONTINUE
      IF(IEVWSW .NE.1) GO TO 89
      WRITE(MG,85)
85    FORMAT(1H1,//    ,* EQUAL INTERVALS*, ////
     1,  10X, 6H NO.    ,10X, 10HMIN. VALUE    ,10X, 10HMAX. VALUE    ,
     2  10X,  10H NUMBER      ,10X, 10H CUML. NO.    )
      DO 86 J = 1,NBOXE
      JJ = J + 1
      NESUM = NESUM + NBE(J)
      WRITE(MG,87) J, BE(J),BE(JJ), NBE(J )      ,NESUM
87    FORMAT( 1H , 10X, 2H (, I2, 2H)   ,10X, F10.0,10X,  F10.0,10X, I10,
     1  10X, I10  )
```

Write number of Doses in Groups.

```
      IF(NESUM .EQ. NSAMP) GO TO 88
86    CONTINUE
88    CONTINUE

89    CONTINUE
      IF(IHSWSW.NE.1) GO TO 97
C     INSERT HISTOGRAM WRITING ROUTINE HERE
      XYZZ = 1.
97    CONTINUE
C*****TEMPORARY*****
      ILSFSW = KTA
      IF(ILSFSW .NE.1 ) GO TO 90
      WRITE(MW,91)
91    FORMAT( 1H1, ///,10X,*FATALITIES,CASUALTIES AND INJURIES FROM INDI
     1VIDUAL SAMPLES* ,//////,1H0,*    NO.     *  ,5X, *FATALITIES*, 5X,
     2 *CASUALITIES*, 5X, * INJURIES *, 5X,*   DOSE    *,//)
      DO 93 J = 1,NSAMP
      TEMP = CSIND(J) - FTIND(J)
      WRITE(MW,92) J, FTIND(J) ,CSIND(J), TEMP,DSEIND(J)
92    FORMAT( 1H , 2X,1H(,I3,1H),8X, F10.0,5X,F10.0,5X,F10.0,5X,F10.0)
93    CONTINUE
90    CONTINUE
101   CONTINUE
      RETURN
      END
```

available in OCPACC version

Detailed Casualty Output

```
      SUBROUTINE WRTLST
```

Final Summary Output

```
      DIMENSION TFAT(100),LFAT(100),TCAS(100),LCAS(100)
      COMMON/RUNSW/ICTYCB,IPRINT,ICLUST,ISINSW,LABEL(20,2),ISPWSW,ILGWSW,
     1IEVWSW,DMOST,NOSEV,IHSWSW,ILVSW,DOSELV(5),IPROSW,PROLV(3),ICATSW,
     2 MDSTP,MLVTP,MMFTP,ICLSW,JREST,JRESTN,JRESTA,JRESTD,JRESTE,JRESTF,
     3 IFRSK,IDMAS,NSIMP,ISTRAT,IWNDST,ISHRSW,IWLVSW,MWCK,ISEED,SEED,
     4 MSDSW,CSEED,IFRCAS,NPFLV,PFLVU(7),PFLVR(7),FDSEM,FDSESG,CDSEM,
     5 CDSESG,MDCAL,ICLYLD,ICLYLG,FRPFU(7) ,FRPFR(7)
     6,ILSTCI, ISTIN(50),ICTIN(100),ISEAS,NSTIN,NCTIN
      COMMON/RUNPR/ KNSAMP,NRPIS,NSMPCT,ILAST,ISINO,ISTLO,JSTCO,JSTNM,
     1 ISTC(51),ISTNM(51),ICARRY(100),DSEPLT,IDSEPT,XPLT,YPLT
     2,DSEPLA,DSEPLB,DSEPLC,ISWING,JSWING,FRFATU,FRCASU,FRFATR,FRCASR
      COMMON/TARPR/ NAMETR(10),TRPT,TRPU,TRPR,TRPUA,IRSTCO,IRCOCO,
     1IRUAC,IREAC,ISRSAC,NOTR,TRCGLA,TRCGLO,YTL,XTL,SIGTRH,SIGTRL,
     2ALTR,NPLCS,NMCOS,NTT(52),FUTT(52),PTTT(52),PRTT(52),PATT(52)
      COMMON/STAPR/DTCT,DMAX,RV(51),BE(502),XLAV(51),NB(51),NBE(501),
     1NBL(50),NBOXA,NBOXL,NBOXE,SMLDOS,SD,SDS,SDC,SDF,SSD,SSDS,SSDC,SSDF
     2,NSML,NBXST(50,52),FBXST(100,52),CBXST(100,52),PROFT(50),PROCS(50)
     3,SFT,SFTS,SFTC,SFTF, SCA, SCAS,SCAC,SCAF,TPOPLV(50,52)
     4,DSEIND(100)
      COMMON/IOPR/ MP,MQ,MS,NA,NC,ND,NE,NF,NG,NH,NI,NJ,NK,NL,NM
      COMMON/LAZY/ITA,ITB,ITC,ITD,ITE,KTA,KTB,KTC,KTD,KTE,KTF,KIG,KTH
      COMMON/LAZYA/ITAA(200),KTAA(200)


      WRITE(MW,15)
15    FORMAT(1H1)
      WRITE(MW,16)
16    FORMAT (/////,30X,*SUMMARY OUTPUT BY STATE* ,///////)
      DO 31 I = 1,ISTNO
      WRITE(MW,15)
      IREP = ICARRY(I)
```

By State

```
      WRITE(MW,35) ISTNM(IREP),ISTC(IREP)
35    FORMAT( 1H0,*SUMMARY FOR THE STATE OF *, A4,   *   WITH STATE CODE
     1 * ,A2)
      WRITE(MW,36) NTT(I),PTTT(I),PUTT(I),PRTT(I),PATT(I)
36    FORMAT(1H0, *  IN *, I3, * COUNTIES POPULATION IS ---  TOTAL = *,
     1 F10.0, *   URBAN = *,F10.0,*  RURAL = * , F10.0,     *   URBANIZED
     2 AREA = *,F10.0)
      IF(IFRCAS .NE.I) GO TO 50
      IG0 = 1
48    CONTINUE
      SUMF = 0.
      SUMFS = 0.
      SUMI = 0.
      SUMIS = 0.
      DO 41 K = 1,100
      TFAT(K) = FBXST(K,I)
      TCAS(K) = CBXST(K,I) - FBXST(K,I)
      SUMF = SUMF + TFAT(K)
      SUMFS = SUMFS + TFAT(K)*TFAT(K)
      SUMI = SUMI + TCAS(K)
      SUMIS = SUMIS + TCAS(K)*TCAS(K)
41    CONTINUE

      CALL FORD(TFAT,LFAT,100)
      CALL FORD(TCAS,LCAS,100)
      XMNF = 0.01*SUMF
      XMNI = 0.01*SUMI
      VARF = .01*SUMFS - XMNF*XMNF
      VARI = 0.01*SUMIS - XMNI*XMNI
      SGF = SQRT(ABS(VARF))
      SGI = SQRT(ABS(VARI))
      ITM = LFAT(1)
      SMLF = TFAT(ITM)
      ITM = LCAS(1)
      SMLI = TCAS(ITM)
      ITM = LFAT(100)
      BIGF = TFAT(ITM)
      ITM  = LCAS(100)
      BIGI = TCAS(ITM)
      WRITE(MW,43)
43    FORMAT(////,1H ,  20X, * FATALITY STATISTICS FROM SHELTERED POPULA
     1TION UNDER FALLOUT RISK *)
      WRITE(MW,42) XMNF,SGF,SMLF,BIGF
42    FORMAT(     1H0, *MEAN NUMBER OF FATALITIES = *,F10.0, *, WITH STD
     1. DEV. = * ,F10.0,*   MINIMUM NUMBER = *,F10.0,* MAXIMUM NUMBER =
     2 *, F10.0)
      WRITE(MW,44)XMNI,SGI,SMLI,BIGI
44    FORMAT(1H0, *MEAN NUMBER OF INJURIES = * ,F10.0, *, WITH STD. DEV.
     1 = *,F10.0, *    MINIMUM NUMBER = *,F10.0, * MAXIMUM NUMBER = *,
     2 F10.0)
      WRITE(MW,47)
47    FORMAT(1H0, 10X, *PERCENTILE*, 10X, *FATALITIES*,10X,* INJURIES *,
     1  //)
      DO 45 J = 1, 9
      JJ = 10*J
      ITM = LFAT(JJ)
      TMP = TFAT(ITM)
      ITM = LCAS(JJ)
      TMPP = TCAS(ITM)
      WRITE(MW,46) JJ,TMP,TMPP
46    FORMAT(1H  ,10X,I10,10X, F10.0,10X,F10.0)
```

Fatality Statistics

```
45      CONTINUE
        WRITE(MG,66)
66      FORMAT( ///  ,      30X,*DOSE STATISTICS FOR SPECIFIED RANGES*,//)
        IF(IGO .EQ.2) GO TO 49
50      CONTINUE
        NSSUM = 0
        WRITE(MG,71)
71      FORMAT( 1H0,                          4X, 6H NO.      4X, 10HMIN. VALU
       1E    , 4X, 10HMAX, VALUE    , 4X,   10H NUMBER     , 4X,
       2  10H CUML. NO. ,4X, 10H PER CENT     ,4X,  10HCUML. PCT.  ,4X ,
       310HTOT. POPN.          )
        ANTOT = NTT(I)*NSAMP
        DO 72 J = 1,NBOXA
        JJ = J + 1
        NSSUM = NSSUM + NBXST(J,I)
        TEM = NBXST(J,I)
        TEM = TEM*100./XNTOT
        TTEM = NSSUM
        TTEM = TTEM*100./XNTOT
        TTTM = TPOPLV(J,I)/XNSAMP
        WRITE(MG,73)  J , BV(J) , BV(JJ), NBXST(J,I),NSSUM,TEM,TTEM,
       , TTTM
73      FORMAT( 1H  ,  4X, 1H( , I2, 1H),  4X, F10.0, 4X, F10.0,  4X,
       1 I10,4X,I10,4X ,F10.3, 4X,F10.3, 4X,F10.0)
72      CONTINUE
31      CONTINUE
        WRITE(MG,40)
40      FORMAT(1H1,//////, 50X,*NATIONWIDE SUMMARY*,///)
        NSSUM = 0
        TTOT = 0.
        NTOT = 0
        RTOT = 0.
        UTOT = 0.
        ATOT = 0.
        DO 61 I = 1,ISTNO
        NTOT = NTOT + NTT(I)
        TTOT = TTOT + PTTT(I)
        RTOT = RTOT + PRTT(I)
        UTOT = UTOT  + PUTT(I)
        ATOT = ATOT + PATT(I)
61      CONTINUE
        ANT = NTOT*NSAMP
        WRITE(MG,62) TTOT,UTOT, RTOT,ATOT
62      FORMAT(  1H0, *NATIONWIDE TOTAL POPULATION = *,F12.0,/, 10X,
       1 *URBAN POPULATION = *,F12.0,/,10X,*RURAL POPULATION = *,F12.0,/,
       2 10X,  *URBANIZED AREA POPULATION = *,F12.0 ,//  )
        IF(IFRCAS .NE.1 ) GO TO 64
        ISTNO = 52
        IGO = 2
        I = 52
        GO TO 48
49      CONTINUE
64      CONTINUE
        WRITE(MG,71)
        DO 75 J = 1,NBOXA
        JJ = J + 1
        NSSUM = NSSUM +  NBXST(J,52)
        TEM = NBXST(J,52)
        TEM = TEM*100./XNT
        TTEM = NSSUM
        TTEM = TTEM*100./XNT
```

Dose statistical

Stat Calculations

Nationwide Calculations

```
      TTTM = IPOPLV(J,52)/XNSAMP
      WRITE(NG,73)  J , BV(J) , BV(JJ), NBXST(J,52),NSSUM ,TEM,TTEM,
     1 TTTM
75    CONTINUE
      RETURN
      END

      SUBROUTINE MONTE


      DIMENSION XWR(500),YWR(500)

      COMMON/RUNSW/ICTYCB,IBINT,ICLUST,ISINGW,LABEL(20,2),ISPWSW,ILGWSW,
     1IEVWSW,DMOST,NDSEV,IHSWSW,ILVSW,DOSELV(5),IPROSW,PROLV(3),ICATSW,
     2 MDSTP,MLVTP,MMPTP,ICLSA,JREST,JRESTN,JRESTA,JRESTD,JRESTE,JRESTF,
     3 IFRSK,IDMAS,NSAMP,ISTRAT,IWNDST,ISHRSW,IWLVSW,MUCK,ISEED,SEED,
     4 MSDSW,CSEED,IFRCAS,NPFLV,PFLVU(7),PFLVR(7),FDSEM,FDSESG,CDSEM,
     5 CDSESG,MDCAL,ICLYLD,ICLYLG,FRPFU(7) ,FRPFR(7)
     6,ILSTCT, ISTIN(50),ICTIN(100),ISEAS,NSTIN,NCTIN
      COMMON/RUNPR/ XNSAMP,NRPTS,NSMPCT,ILAST,ISTNO,ISTLO,JSTCO,JSTNM,
     1 ISTC(51),ISTNM(51),ICARRY(100),DSEPLT,IDSEPT,XPLT,YPLT
     2,DSEPLA,DSEPLB,DSEPLC,ISWING,JSWING,FRFATU,FRCASU,FRFATR,FRCASR
      COMMON/CLSPR/ YLOC(500),FISSC(500),XWC(500),YWC(500),SIGXC(500),
     1 SIGYC(500),NCLS,SARY(500,5),YLDI(7),YLDM(6),DARYL(7,19),
     2ARRY(7,40)
      COMMON/TARPR/ NAMETR(10),TRPT,TRPU,TRPR,TRPUA,IRSTCO,IRCOCO,
     1IRUAC,IREAC,ISMSAC,NUTR,TRCGLA,TRCGLO,YTL,XTL,SIGTRH,SIGTRL,
     2ALTR,NPLCS,NMCDS,NTT(52),PUTT(52),PTTT(52),PRTT(52),PATT(52)
      COMMON/WNDPR/WIND,W,SGWP,SGW,SHRST,SHR,SGSHRP,SGSHR,WMNDR,ALWND,
     1 CAL,SAL, FLATMO(100),FLONMO(100),DEGMO(100),SPDMO(100),VSMO(100),
     2 NUMO
      COMMON/STAPR/DTOT,DMAX,BV(51),BE(502),XLBV(51),NB(51),NBE(501),
     1NBL(50),NBOXA,NBOXL,NBOXE,SMLDOS,SD,SDS,SDC,SDF,SSD,SSDS,SSDC,SSDF
     2,NSML,NBXST(50,52),FBXST(100,52),CBXST(100,52),PROFT(50),PROCS(50)
     3,SFT,SFTS,SFTC,SFTF, SCA, SCAS,SCAC,SCAF,IPOPLV(50,52)
     4,DSEIND(100)
      COMMON/CLFLPR/YIELD,FISS, V  ,SHR,XL,SIGW,XDW,YCW,DSSS,STR(5)
      COMMON/GWSEG/ YIELN,FISN,HOB,Z,SHZ,XDW,YCN,XTBA,GARRY(19)
      COMMON/LAZY/ITA,ITB,ITC,ITD,ITE,KTA,KTB,KTC,KTD,KTE,KTF,KTG,KTH
      COMMON/LAZYA/ITAA(200),KTAM(200)
      DATA ( DELTH = .62831853),(TORAD = 0.01745329)


C ***********3600/6400 COMPATABILITY CHANGE ***************
C       IRAN = -1
        IRAN = 0
        NSMPCT = 0
30      CONTINUE
C       COMPUTE RELATIVE WEAPON LOCATIONS
        DO 12 I = 1,NCLS
        XWR(I) = XTL - XWC(I)
        YWR(I) = YTL - YWC(I)
12      CONTINUE
        IF(MSDSW .NE.1) GO TO 15
C ***********3600/6400 COMPATABILITY CHANGE ***************
C       CALL RANFSET(CSEED)
        CALL RANSET(CSEED)
15      CONTINUE
```

Generate wind statistics

```
C      PICK A WIND
       IF (ISTRAT .EQ.1) GO TO 53
       NSMPCT = NSMPCT + 1
       TEMP = RANF(IRAN)
       THETA=6.2831853 *TEMP
       TEMP = RANF(IRAN)
       TMZ = TEMP
       GO TO 54
53     CONTINUE
       POLD = 0.
       DO 31 IS= 1,10
       THOLD = 0.
       DO 32 JS= 1,10
       DO 33 KS= 1,NRPTS
       NSMPCT = NSMPCT + 1
       TEMP = RANF(IRAN)
       THETA=THOLD+DELTH*TEMP
       TEMP = RANF(IRAN)
       TMZ=POLD+0.1*TEMP
54     CONTINUE
       VWRS = 2.*SGW*SGW*ALOG(1./(1.-TMZ))
       VWR = SQRT(VWRS)
       WINDS = WIND*WIND
       VSQ = WINDS + VWRS - 2.*WIND*VWR*COS(THETA)
       V = SQRT(VSQ)
       SAL = VWR* SIN(THETA)/V
C ***********3600/6400 COMPATABILITY CHANGE **************
       ALPH=ASIN (SAL)
C      ALPH=ASINF(SAL)
C      THETA IS IN RADIANS CLOCKWISE FROM WEST. ANGLE BETWEEN WIND, VWR
C      ALPHA IS ANGLE BETWEEN V AND WIND
C      AL IS IN RADIANS COUNTERCLOCKWISE FROM EAST
C      WMNDR IS IN DEGREES CLOCKWISE FROM NORTH.
C      EG. A WIND FROM WEST TO EAST HAS AL= 0 AND WMNDR = 270
       CALSG = VSQ + WINDS - VWRS
       IF(CALSG .GE. 0.) GO TO 44
       ALPH = 3.14159265 - ALPH
       GO TO 45
44     CONTINUE
       IF(SAL .LT. 0.) ALPH = 6.2831850 + ALPH
45     CONTINUE
       AL = ALPH + 4.71238897 - WMNDR*TORAD
       SAL=SIN(AL)
       SALS = SAL * SAL
       CAL = COS(AL)
       CALS = CAL*CAL

C      NOW PICK A SHEAR

       IF(SGSHRP .EQ. 0.)GO TO 42
       SUM = 0.
       DO 43 J = 1,12
       TEMP = RANF(IRAN)
       SUM = SUM + TEMP
 43    CONTINUE
       SHR = SHRST +  SGSHR*(SUM - 6.)/12.
       IF(SHR .LT. 0.) SHR = - SHR
 42    CONTINUE

C      SET UP SCREENING FOR THIS WIND
       IF(V .LT.10.) GO TO 61
```

Wind Statistics

```
      ISLOW = 0
      IF(SHR .GT. 0.3) GO TO 62
      CSCRN = 175.
      GO TO  66
62    CONTINUE
      CSCRN = 250.
      GO TO 66
61    CONTINUE
      IF(V .LT.5.) GO TO 64
      ISLOW = 1
      CSCRN = 250.
      GO TO 66
64    CONTINUE
      ISLOW = 1
      CSCRN = 300.
      DSCRN = 300.
66    CONTINUE
      MQ = 6LOUTPUT
      ALD = AL/TORAD
      THD = THETA/TORAD
      WRITE(MQ,630) NSMPCT,AL,V,SHR,TMZ,VWR,THETA,ALPH,THD,ALD
630   FORMAT(1H0, * N-AL-V-SHR-TMZ-VWR-TH-ALPH-THD-ALD =*,I5,9F8.3)

      DTOT = 0.
C     NOW COMPUTE DOSES FOR EACH WEAPON
      DO 71 K = 1,NCLS
C     SCREENING CHECK
```

Step through clusters

```
      IF(ISLOW .NE.0) GO TO 72
      YCW = -XWR(K)*SAL + YWR(K)*CAL
      IF(ABS(YCW) .GT. CSCRN) GO TO 71
      XDW = XWR(K)*CAL + YWR(K)*SAL
      IF(XDW .LT. -80.) GO TO 71
      XL = 2.*(SIGAC(K)*CALS + SIGYC(K)*SALS) + 5.
      IF(XDW .LT.-XL) GO TO 71
      SIGW = 1.414*(SIGYC(K)*CALS + SIGAC(K)*SALS) + 5.
      DO 74  J = 1,5
```

Screening Test Passed

```
      STR(J) = SARY(K,J)
74    CONTINUE
      CALL CFALWD
      DTOT = DTOT + DSSS
      WRITE(MQ,640) K ,XWC(K),YWC(K),XWR(K),YWR(K),YLDC(K),FISSC(K),
     1 XDW,YCW,XL,SIGW, DSSS
640   FORMAT(1H , I5, 11F11.3)
      GO TO 71
```

Temporary to print detailed statistics

```
72    CONTINUE
      IF(ABS(XWR(K)) .GT. DSCRN) GO TO 71
      IF(ABS(YWR(K)) .GT. CSCRN) GO TO 71
      XDW = XWR(K)*CAL + YWR(K)*SAL
      YCW = -XWR(K)*SAL + YWR(K)*CAL
C     CHOOSE SIZE CLASS HERE
      DO 75 I = 1,6
      II = I
      IF(YLDC(K)    .LT.YLOM(I))GO TO 76
75    CONTINUE
      II = II + 1
76    CONTINUE
      DO 73 I = 1,19
      QARRY(I) = QARYL(II,I)
73    CONTINUE
      ATHA = ABS(     SIGXC(K)*SIGYC(K))
      HOB = 0.
      Z = V
      SHZ = SHR
      XDN = XDW
      YCN = YCW
      CALL QFALFW
      CALL QFALDW
      CALL QFALCW
      DTOT = DTOT + QARRY(13)*FISSC(K)
C     ASSUMING FALLYQ ORIGINALLY CALLED WITH FISS OF 1
      TTMP =  QARRY(13)*FISSC(K)
      WRITE(MQ,640) K ,XWC(K),YWC(K),XWR(K),YWR(K),YLDC(K),FISSC(K),
     1 XDW,YCW,XL,SIGW, TTMP
71    CONTINUE
      IF(DTOT.LT.0.00001) DTOT=0.0
      CALL BOXFL
      IF(ISTRAT .EQ. 1) GO TO 55
      IF(NSMPCT .LT. NSAMP) GO TO 30
      GO TO 56
55    CONTINUE
33    CONTINUE
      THOLD = THOLD + DELTH
32    CONTINUE
      POLD = POLD + 0.1
31    CONTINUE
56    CONTINUE
      RETURN
      END
```

Detailed Temporary Output

```
      SUBROUTINE PREPAR
```

Initialization for each Target

```
      COMMON/STAPR/DTOT,DMAX,BV(51),BE(502),XLBV(51),NB(51),NBE(501),
     1NBL(50),NBOXA,NBOXL,NBOXE,SMLDOS,SD,SDS,SDC,SDF,SSD,SSDS,SSDC,SSDF
     2,NSML,NBXST(50,52),FBXST(100,52),CBXST(100,52),PROFT(50),PROCS(50)
     3,SFT,SFIS,SFTC,SFTF, SCA, SCAS,SCAC,SCAF,TPUPLV(50,52)
     4,DSEIND(100)

      DO 51 J = 1,NBOXA
      NB(J) = 0
51    CONTINUE
      DO 52 J = 1,NBOXL
      NBL(J) = 0
52    CONTINUE
      DO 53 J = 1,NBOXE
      NBE(J) = 0
53    CONTINUE
      NB(NBOXA+1) = 0
      NBL(NBOXL+1) = 0
      NBE(NBOXE+1) = 0
      DMAX = 0.
      SD = 0.
      SDS = 0.
      SDC = 0.
      SDF = 0.
      SSD = 0.
      SSDS = 0.
      SSDC = 0.
      SSDF = 0.
      NSML = 0
      RETURN
      END
```

Nevans Standard

```
      FUNCTION CUMNOR(TA)
C     APPROX HASTINGS P.187 FOR CUM NOR.
      IF(TA.LT.5.) GO TO 14
      CUMNOR = 1.0
      RETURN
   14 CONTINUE
      IF(TA .GT.-5.) GO TO 16
      CUMNOR = 0.0
      RETURN
16    CONTINUE
      TM = ABS(TA/1.414213562)
      TMP =1.+TM*(0.0705230784+TM*(0.0422820123+TM*(0.0092705272+TM*
     1(0.0001520143+TM*(0.0002765672+TM*0.0000430638)))))
      TMP=TMP*TMP
      TMP=TMP*TMP
      TMP=TMP*TMP
      TMP=TMP*TMP
      CUP= 1.-1./TMP
      IF(TA .LT. 0.)GO TO 21
      CUMNOR = 0.5*(1. + CUP)
      RETURN
21    CONTINUE
      CUMNOR = 0.5*(1. - CUP)
      RETURN
      END
```

Nevans Standard

```
      SUBROUTINE FORD(ARR,LARR,NITM)
      DIMENSION ARR(101),LARR(101)
      IT = 1
      LARR(IT) = 1
      VAL = ARR(IT)
10    CONTINUE
      IT = IT + 1
      IF(IT .GT. NITM) GO TO 100
      IF(ARR(IT) .LT. VAL) GO TO 12
      LARR(IT) = IT
      VAL = ARR(IT)
      GO TO 10
12    CONTINUE
      JK = IT - 1
      DO 13 J = 1,JK
      JJ = J
      ITM = LARR(J)
      IF(ARR(ITM).LT. ARR(IT)) GO TO 13
      GO TO 14
13    CONTINUE
14    CONTINUE
      IK = IT - JJ
      DO 15 K = 1,IK
      KK = IT - K
      KKK = KK + 1
      LARR(KKK) = LARR(KK)
```

H1

```
15     CONTINUE
       LARR(JJ) = IT
       GO TO 10
100    CONTINUE
       RETURN
       END

       SUBROUTINE CFALLY
```

nevern Standard

```
C      CALCULATES FALLOUT DOSES FROM CLUSTERS OF WEAPONS USING A
C      SIMPLFIED WSEG 10 FALLOUT MODEL.
C      XL IS THE HALF LENGTH OF THE CLUSTER IN THE DOWNWIND DIRECTION
C      SIGW IS STD. DEV. OF CLUSTER IN TH E CROSSWIND DIRECTION.
C      DWN DOWNWIND DIRECTION,CRS CROSSWIND DIRECTION.
C      STR IS USED TO STORE YIELD DEPENDENT PARAMETERS
C      STR(1) IS CHARACTERISTIC TIME T.
C      STR(2)  IS 2.71/T**1.382
C      STR(3)  IS A USED TO CALCULATE SHEAR SIGMA
C      STR(4) IS B USED TO CALCULATE SHEAR SIGMA
C      STR(5) IS PRODUCT OF YIELD TIMES FISSION FRACTION TIMES K.
C      ASSUMES 0 HEIGHT OF BURST SO ANY CORRECTION MUST BE EXTERNAL.

       COMMON/CLFLPR/YIELD,FISS,WIND,SHR,XL,SIGW,DWN,CRS,DOSE,STR(5)


       ALT = ALOG10(YIELD)
       STR(1) = 7.5 + 1.66*ALT
       STR(2) = 2.71/(STR(1)**1.382)
       STR(3) = 2. + 3.*ALT
       STR(4) = 7.5 + 1.5*ALT
       STR(5) = 2000000.*YIELD*FISS
       RETURN
       END

       SUBROUTINE CFALWD
```

nevern Standard

```
C      FOR WEAPONS IN CLUSTERS
C      DOES BOTH WIND DEPENDENT AND DISTANCE DEPENDENT CALCULATIONS.
C      ASSUMES CFALLY HAS BEEN CALLED TO FILL STR
C      CLUMSY WRITING USED TO ATTEMPT TO SPEED CALCULATION.

       COMMON/CLFLPR/YIELD,FISS,WIND,SHR,XL,SIGW,DWN,CRS,DOSE,STR(5)


       SIGC = STR(3) + STR(4)*(DWN + XL)*SHR/WIND
       IF(DWN .LT.XL) GO TO 21
C      HERE WE ARE BEYOND THE CLUSTER
       RAT = DWN/(WIND*STR(1))
       IF(RAT .LT. 0.5) GO TO 11
       IF(RAT.GT.8.)GO TO 23
C      USE POLYNOMIAL APPROXIMATION HERE.
       FAC = 0.85419 + RAT*(0.421 + RAT*(-0.019286 + RAT*0.00929))
       FACS = FAC*FAC
       FD = STR(2)/(WIND*FACS*FACS)
12     CONTINUE
       IF(DWN .LT. 5.*SIGW) GO TO 15
C      SOMETHING LIKE DWN .LT. 75. COULD ALSO BE USED HERE
       SIGUS = SIGC*SIGC + SIGW*SIGW
C      SIGUS IS SHR SIGMA MODIFIED FOR CLUSTER SIZE FOR DOWNWIND APPRX.
```

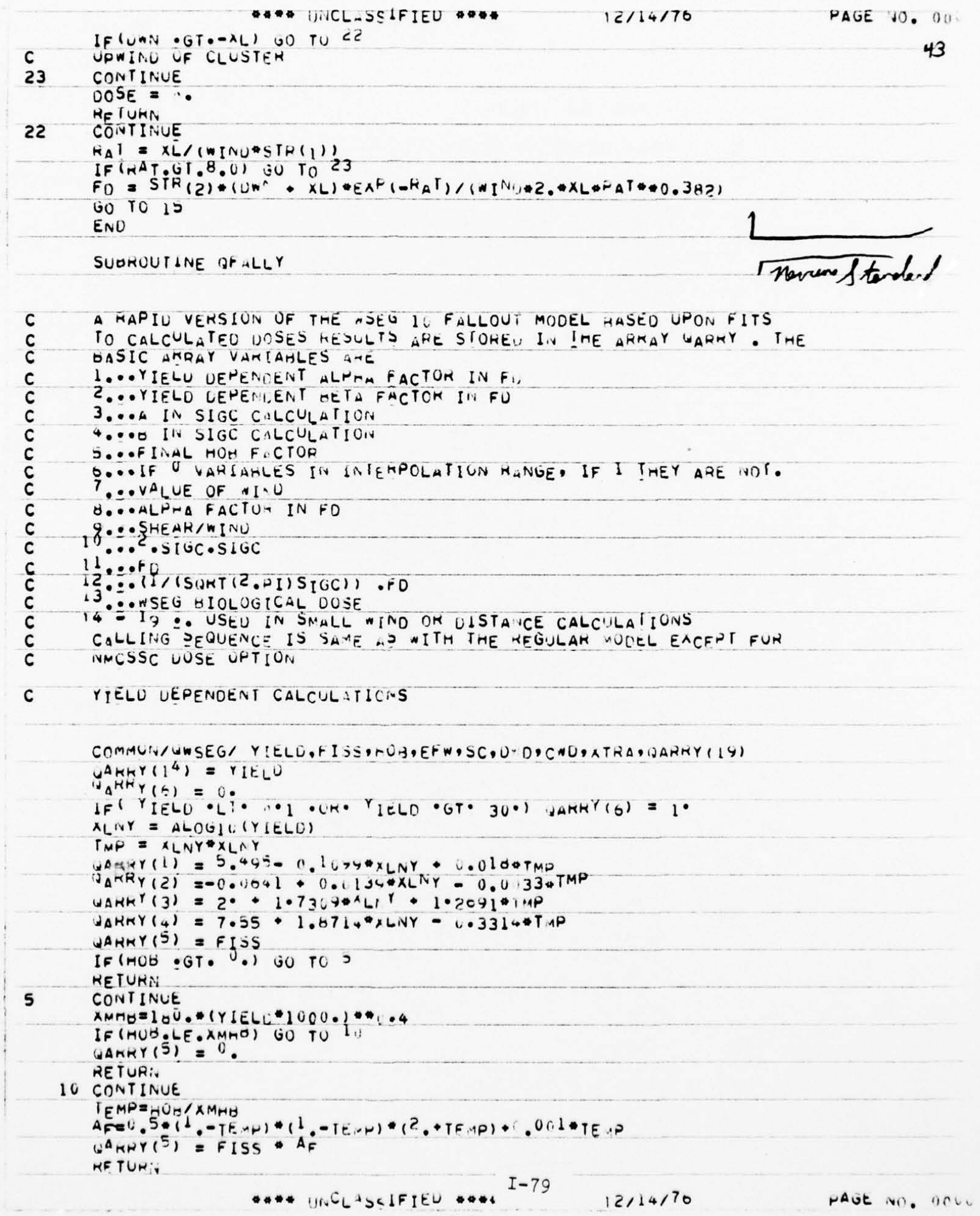
```
      SIGU = SQRT(SIGUS)
      TMP = 0.5*CRS*CRS/SIGUS
      IF(TMP .GT. 6.) GO TO 23
      FC = 0.39894228*EXP(-TMP) /SIGU
18    CONTINUE
      DOSE = STR(5)*FD*FC
      RETURN

15    CONTINUE


C     COMPUTE CROSSWIND FACTOR HERE BY DIFFERENCE OF TWO CUMULATIVE
C     NORMALS. CUMNOR APPROXIMATION DIRECT FROM HASTINGS.
      R = 1.414*SIGW
      TMP1 = ABS(CRS)/SIGC
      TMP2 = R/SIGC
      X = TMP1 + TMP2
      XX= TMP1 - TMP2
      IF(XX .GT. 6.) GO TO 23
       IF(X.LT.4.0) GO TO 31
       IF(XX.LT.4.0) GO TO 32
      FC=0.0
      GO TO 18
 32     CONTINUE
      IF(XX.GT.-4.0) GO TO 33
      FC=0.5/R
      GO TO 18
 33   CONTINUE
      CMN 1=1.0
 31    CONTINUE
      FAC = 1. + X*(0.278393+X*(0.230389+X*(0.00972+X* 0.078108)))
      FACS = FAC * FAC
      CMN1 = 1. - 0.5/(FACS*FACS)
      IF(XX.LT.4.0) GO TO 35
      CMN2=1.
      GO TO 17
 35   CONTINUE
      IF(XX.GT.-4.0)    GO TO 36
      CMN2=0.0
      GO TO 17
 36   CONTINUE
      X = ABS(XX)
      FAC = 1. + X*(0.278393+X*(0.230389+X*(0.00972+X* 0.078108)))
      FACS = FAC * FAC
      IF(XX.LT. 0.) GO TO 16
      CMN2 = 1. - 0.5/(FACS*FACS)
      GO TO 17
16    CONTINUE
      CMN2 = 0.5/(FACS*FACS)
17    CONTINUE
      FC = 0.5*(CMN1 - CMN2)/R
      GO TO 18

11    CONTINUE
C     USE EXPLICIT CALCULATION RATHER THAN APPROXIMATION BELOW
      FD = STR(2) *EXP(-RAT)/(WIND*RAT**0.382)
      GO TO 12


21    CONTINUE
C     HERE WE ARE IN THE CLUSTER AND USE LINEAR VARIATION OF FD ALONG IT
```



```
      IF(DWN .GT.-XL) GO TO 22
C     UPWIND OF CLUSTER
23    CONTINUE
      DOSE = 0.
      RETURN
22    CONTINUE
      RAT = XL/(WIND*STR(1))
      IF(RAT.GT.8.0) GO TO 23
      FD = STR(2)*(DWN + XL)*EXP(-RAT)/(WIND*2.*XL*RAT**0.382)
      GO TO 15
      END

      SUBROUTINE QFALLY
```

Nevrue Standard

```
C     A RAPID VERSION OF THE WSEG 10 FALLOUT MODEL BASED UPON FITS
C     TO CALCULATED DOSES RESULTS ARE STORED IN THE ARRAY QARRY . THE
C     BASIC ARRAY VARIABLES ARE
C     1...YIELD DEPENDENT ALPHA FACTOR IN FD
C     2...YIELD DEPENDENT BETA FACTOR IN FD
C     3...A IN SIGC CALCULATION
C     4...B IN SIGC CALCULATION
C     5...FINAL HOB FACTOR
C     6...IF 0 VARIABLES IN INTERPOLATION RANGE, IF 1 THEY ARE NOT.
C     7...VALUE OF WIND
C     8...ALPHA FACTOR IN FD
C     9...SHEAR/WIND
C    10...2.SIGC.SIGC
C    11...FD
C    12...(1/(SQRT(2.PI)SIGC)) .FD
C    13...WSEG BIOLOGICAL DOSE
C    14 = 19 .. USED IN SMALL WIND OR DISTANCE CALCULATIONS
C     CALLING SEQUENCE IS SAME AS WITH THE REGULAR MODEL EXCEPT FOR
C     NMCSSC DOSE OPTION

C     YIELD DEPENDENT CALCULATIONS

      COMMON/QWSEG/ YIELD,FISS,HOB,EFW,SC,D-D,CWD,XTRA,QARRY(19)
      QARRY(14) = YIELD
      QARRY(6) = 0.
      IF( YIELD .LT. 0.1 .OR. YIELD .GT. 30.) QARRY(6) = 1.
      XLNY = ALOG10(YIELD)
      TMP = XLNY*XLNY
      QARRY(1) = 5.495- 0.1079*XLNY + 0.018*TMP
      QARRY(2) =-0.0641 + 0.0134*XLNY - 0.0033*TMP
      QARRY(3) = 2. + 1.7309*XLNY + 1.2091*TMP
      QARRY(4) = 7.55 + 1.8714*XLNY - 0.3314*TMP
      QARRY(5) = FISS
      IF(HOB .GT. 0.) GO TO 5
      RETURN
5     CONTINUE
      XMHB=180.*(YIELD*1000.)**0.4
      IF(HOB.LE.XMHB) GO TO 10
      QARRY(5) = 0.
      RETURN
   10 CONTINUE
      TEMP=HOB/XMHB
      AF=0.5*(1.-TEMP)*(1.-TEMP)*(2.+TEMP)+0.001*TEMP
      QARRY(5) = FISS * AF
      RETURN
      END

      SUBROUTINE QFALFW
C     WIND DEPENDENT CALCULATIONS
      COMMON/QWSEG/ YIELD,FISS,HOB,EFW,SC,DWD,CWD,XTRA,QARRY(19)
      QARRY(7) = EFW
      IF(EFW.LT. 3.) GO TO 5
      QARRY(9) = SC/EFW
      QARRY(8) = QARRY(1) - 0.995*ALOG10(EFW)
      RETURN
5     CONTINUE
      QARRY(9) = SC
      QARRY(8) = 2.
      WHS = 1. - 0.5*EFW
      IF(WHS .LT. 0.) WHS = 0.
      QARRY(15) = (4.545 - 0.0745*QARRY(14)) + (0.1222 + 0.0078*
     1 QARRY(14) )*WHS - (1.2222 + 0.0278*QARRY(14) ) *SC
      QARRY(16) = -0.0486 + 0.00316*QARRY(14)
      QARRY(17)=( 0.2444 - 0.02444*QARRY(14) -(0.8977+ 0.1323*QARRY(14))
     1 *WHS )*0.001
      QARRY(18) = 3.14 + 0.51*QARRY(14) - (0.33+ 0.03*QARRY(14))*EFW
     1 + (42.35+ (-19.0975 + 0.4225*QARRY(14))*EFW)*SC
     2 + (69. + (-27.35  + 1.15 *QARRY(14))*EFW)*SC*SC
      QARRY(19) = (3.611 + 0.039*QARRY(14) )*SC
      RETURN
      END

      SUBROUTINE QFALDW
C     DOWNWIND DISTANCE DEPENDENT CALCULATIONS
      COMMON/GWSEG/ YIELD,FISS,HOB,EFW,SC,DWD,CWD,XTRA,QARRY(19)
      IF( QARRY(7) .LT. 3.) GO TO 20
      XSCL = DWD/QARRY(7)
      SIGC = QARRY(3) + QARRY(4)*QARRY(9)*DWD
      SIGC = SQRT(SIGC*SIGC + XTRA)
      IF(XSCL .LT . 1.) GO TO 5
      XLGFD = QARRY(8) + QARRY(2)*XSCL
      TUSE = 2.5066*SIGC
      IF(XSCL .GT. 15.6) GO TO 10
      TMP = XSCL - 15.6
      XLGFD = XLGFD + 0.0015*TMP*TMP
10    CONTINUE
      QARRY(10) = 2.*SIGC*SIGC
      IF( XLGFD .LT. -8.) XLGFD = -8.
      IF(XLGFD .GT.10) XLGFD = 10.
      QARRY(11) = QARRY(5) *(10.**XLGFD)*QARRY(14)
      QARRY(12) = QARRY(11)/TUSE
      RETURN
5     CONTINUE
      ALW = ALOG10(QARRY(7)/20.)
      DELTN = 3. + 5.5*XLW
      FACT = 2. - ALOG10(QARRY(14))
      DELTA = DELTN*FACT
      XLDMXN = 3.355 - 0.306*XLW - 0.275*QARRY(9)/QARRY(7)
      XLDMX = XLDMXN + 0.443*(FACT - 1.)
      DWDS = DWD*FACT
      IF ( DWDS .LT. DELTA) GO TO 6
      XLGFD = XLDMA
      TUSE = 1.
      GO TO 10
6     CONTINUE
      XLGFD = XLDMA- (DWDS-DELTA)*(DWDS-DELTA)*0.0179
      TUSE = 1.
      GO TO 10
20    CONTINUE
      IF( DWD .GT.0. )GO TO 21
      XLY = ALOG10(QARRY(14))
      XLDMA = 4.35 - 0.56*XLY        - 0.12*QARRY(7)- .15*QARRY( 9)
      BOT = 67. + 257.*ALOG10(QARRY(14)   )
      DUSE = 0.5*QARRY(7) - DWD
      XLGFD = XLDMA - DUSE*DUSE/BOT
      GO TO 22
21    CONTINUE
      XLGFD = QARRY(15) + QARRY(16)*DWD + QARRY(17)*DWD*DWD
      IF(QARRY(17) .LT. 1.E-6) GO TO 22
      BETA = -1.*QARRY(16)/(2.*QARRY(17))
      IF( DWD .GT. BETA) XLGFD = -8.
22    CONTINUE
      SIGC = QARRY(18) + QARRY(19)*DWD
      SIGC = SQRT(SIGC*SIGC + XTRA)
      TUSE = 1.
      GO TO 10
      END

      SUBROUTINE QFALCW
C     CROSSWIND DISTANCE DEPENDENT CALCULATIONS

      COMMON/QWSEG/ YIELD,FISS,FOB,EFW,SC,DWD,CWD,XTRA,QARRY(19)
      TMP = CWD*CWD/QARRY(10)
      IF(TMP .GT.10.) TMP = 10.
      QARRY(13) = QARRY(12)*EXP(-TMP)
      RETURN
      END
```

nevuno f Standard

COMPUTER PROGRAM DESCRIPTION

NAME: AGZSEL

SYNOPSIS: Detailed Urban Tract Damage Assessment

TYPE: Semi-Production

USE: Urban Damage Assessment Under a Variety of Conditions. Prime Data Source for Computer.

BACKGROUND: Developed from IDA Program DGZSEL. Variations added to Study Varying Blast Shelter Options and Compare MONTE CARLO with expected Damage Results.

DESCRIPTION: Performs Weapon Optimization and Detailed Damage Assessment. Allows for Several Assessment Functions Reflecting Various Weapons.

INPUT: Parameter Cards in Source Program.
Tract Locations.

OUTPUT: Separate set of Printouts, Printout Maps, etc.
300 Runs
IDA Run 77

STORAGE: IDA Card Deck #84

DOCUMENTATION: IDA Paper P-752, "A Sensitivity Analysis of Urban Blast Fatality Calculations," Leo A. Schmidt, Jan. 1971.

LANGUAGE/SYSTEM: FTN/1604 Operating System

COMMENTS: 1604 Implementation Only. Contains Many 1604 Unique Instructions.

Preceding Page Blank - NOT FILMED

COMPUTER PROGRAM DESCRIPTION

NAME: LANCET

SYNOPSIS: Convert ADAGIO Output to ANCET Input

TYPE: PRODUCTION

USE: To Use ANCET Damage Assessment Calculations on Evacuated Population Data Base

BACKGROUND: Developed as an auxiliary program for MEVUNS system

DESCRIPTION: Converts ADAGIO output data to format needed by ANCET

INPUT: ADAGIO output tape (RSAC ADAGIO)

OUTPUT: ANCET input tape

STORAGE: IDA Card Deck 109

DOCUMENTATION: IDA Study S-394, "Methodology for Evaluating the Vulnerability of National Systems," Vol. I, Part I, Description of Methodology.

LANGUAGE/SYSTEM: Run (FORTRAN)/6400 SCOPE

COMMENTS:

COMPUTER PROGRAM DESCRIPTION

NAME: AIP/ANCET

SYNOPSIS: Nationwide Damage Assessment Calculation

TYPE: PRODUCTION

USE: Nationwide Damage Assessment for Blast and Fallout Given on Attack and Fallout Shelter Assumptions

BACKGROUND: Originally developed at Research Triangle Institute and sent to NCDCF. The version was modified at IDA to increase allowable attack size from 1000 to 3600 weapons. It was used in IDA MEVUNS Study and sent to DCPACC.

DESCRIPTION: Analytic calculation of blast damage to Gaussian distributed population nodes. WSEG10 fallout calculations.

INPUT: Population Data
Attack data
Shelter assumptions

OUTPUT: See IDA Card Deck #175
IDA Run 81, 82, 101, 115

STORAGE: IDA Card Deck #106

DOCUMENTATION: "An Analytical Technique for Urban Casualty Estimation from Multiple Nuclear Weapons," Jeffrey J. Hunter, Operations Research, 15, 1967, pp. 1096-1108: (continued below)

LANGUAGE/SYSTEM: FTN/6400 SCOPE
FTN/3600 SCOPE

COMMENTS: Central Program of the MEVUNS system. Related programs are GCM, ALLEGRO, TOGUM. No modifications to IDA program made since deck sent to DCPACC in 1972. Current IDA deck if of archival interest only since extensive improvements have since been made at DCPACC.

DOCUMENTATION (continued)

"ANCET Improvements, Final Report, Vol. I," Mary B. Woodside, Research Triangle Institute, November 1968 (includes a comprehensive bibliography of Relevant RTI documents); IDA Report S-394, "Methodologies for Evaluating the Vulnerabilities of National Systems," Vol. I, Part I, Description of Methodologies, November 1, 1971

COMPUTER PROGRAM DESCRIPTION

NAME: ALLEGRO

SYNOPSIS: ATTACK GENERATOR

TYPE: PRODUCTION

USE: Nationwide Attack Generator

BACKGROUND: Developed as a rapid attack generator and blast damage assessment routine, later modified to include attack on an economic data base and with ABM defenses, and incorporated with the MEVUNS system.

DESCRIPTION: Blast only optimization based on LaGrange Multipliers and Square Root Damage Low Economic Optimization based on County Resolution data. Attack weight for population and each economic sector are input.

INPUT: Population Data Base
Economic Data Base

OUTPUT: Punched Cards for GEM input
Separate set of 100 runs used for MEVUNS Vol I, Part II Study
IDA Run #71, 75, 83

STORAGE: IDA Card Deck #91

DOCUMENTATION: IDA Study S-394, "Metholodogies for Evaluating the Vulnerability of National Systems," Vol. I, Part I Description of Methodologies, J. McGill, et.al., November 1, 1971.

LANGUAGE/SYSTEM: FTN/6400 SCOPE
FTN/3600 SCOPE

COMMENTS: Available population input tapes based on 1960 census.

COMPUTER PROGRAM DESCRIPTION

NAME: ANDANTE

SYNOPSIS: Detailed Urban Tract Damage Assessment

TYPE: PRODUCTION

USE: Study Combined Blast-INR Effects

BACKGROUND: Developed from Program AGZSEL to include INR inputs and to give output emphasizing combined effects.

DESCRIPTION: Tract population and location data is input. The blast over pressure, INR dose and fatalities and casualties on each tract are computed and summarized. Attack optimization option available.

INPUT: Parameters
Tract Location

OUTPUT: Separate set of printouts
IDA Printout #96, 119

STORAGE: IDA Card Deck 99

DOCUMENTATION: Attached sheets

LANGUAGE/SYSTEM: FTN/6400 SCOPE
FTN/3600 SCOPE
FTN/IBM COS

COMMENTS: The version has old blast and INR input routine. Current DCPACC version has same basic bookkeeping but new blast and INR routine.

The majority of subroutines in the program ANDANTE are NEVUNS Standard and are described in Chapter III. The parameter input stream is spread somewhat in the program and so will be described below.

Card #	Calling Subroutine	Format	Variable	Use
1	RUNPAR	20A4	LABEL(I), I=1,20	Label for run printed on output.
2		I10	JRAD	0 - Do weapon optimization 1 - Do Blast-INR evaluation.
3		A4,6X,I10	NNEND	Name of last city evaluated-stop flag.
			IPUNCH	If 1, punch card as output from optimization.
4		I10,F10.5	JPKTP	1, use subroutine FLPKHU for probability distance 0, use subroutine FLPKA for probability distance
			ADJSTF	Fraction change of mean lethal radius (see FLPKA descriptio
5		I10	LSTAPE	If 1, write output summary as Magnetic Tape in addition to standard output.
6	RADCAL	I10	LSTC	This parameter controls additional reading for the next case to be run. Values are 0 - Read new city and weapon locations 1 - Don't read city or weapo locations 2 - Read new weapon locations only 3 - Do not read new parameters; change parameters by subroutine CHANGE as described below.
7		unused		Describe input on following card.
8		4F10.5,F10.0,F10.5	YIELD	Weapon Yield--MT
			CEP	Weapon CEP, n.m. if negative 10 psi optimized burst altitude
			PSI	Mean lethal overpressure, if negative surface burst. If both PSI and CEP positive, everywhere optimum airburst pressures

Card #	Calling Subroutine	Format	Variable	Use
			DEL	Weapon Reliability
			DESMX	Return per weapon end for optimization
			FMAXWP	Number of weapon end for optimization. Either return for weapon less than DESMX or number of weapons greater than FMAXWP will terminate the optimization.
9		I10	NSP	Number of tracts output in TRDCL
10		unused		Describes input on following card
11		8F10.4	D50	Mean lethal dose, rads
			SIG50	Standard deviation of mean lethal dose, rads
			PF	Initial nuclear radiation protection factor
			C50	Mean casualty dose, rads
			SIGC50	Standard deviation of mean casualty dose
			PSINJ	Mean casualty overpressure
			SIGBC	Standard deviation of PSIINJ as a fraction of RSINJ
			SIGBL	Standard deviation of PSI as a fraction of PSI
12		3F20.6	AR BR CR	INR radiation constants in expression $\ln R = A+B\ln P+C(\ln P)^2$

If a city is to be read:

Card #	Calling Subroutine	Format	Variable	Use
1N	NEWCTY	15A4	NAMEC(I),I=1,15	City name; if the same as NNEND terminate run
2N •		2F10.3,F10.0	Y(I)	Tract latitude (if < 0 terminate read)
• •			X(I)	Tract longitude
•			POP(I)	Tract population

If a set of weapon cards is to be read:

Card #	Calling Subroutine	Format	Variable	Use
1W	PAYIN	55X,I5	IWP	Number of weapons to be read
2W		I5,4F15.6	J	Weapon number
• • • •			PAYZT(J)	Cumulative fatalities computed during optimization to the weapon
•			PAY(Z)	Fatalities computed for the weapo
•			XZ(J)	Weapon Longitude
•			YZ(J)	Weapon Latitude

After this input the read process returns to card 6.

The subroutine change is used to change one or two variables as part of a string of runs. For each change a card is read, under format F10.0,2I2, which defines the variables VARB, IBLK, ILST. VARB is the new value of the variable to be changed. The common block containing this variable is specified by the variable IBLK, and the location of the variable in the common block by the variable ILST. The values of IBLK and ILST to change a particular variable are readily found by reading the subroutine.

The flow of calculations is clear from the program. The first step is reading and documenting of input data. Then the subroutines FLPKHU or FLPKA are used to obtain probability of kill as a function of distance squared. The results of these calculations are placed in the common block /PKPR/. For optimization the calculation is done by the subroutine OPTWPN. For evaluation the calculations for a single tract are accomplished by the subroutine TRDCL and for the entire city by the subroutine RADKL. The input output subroutines and control subroutines are quite transparent. The other subroutines are all described in Chapter IV.

The current source deck for the program at IDA includes a number of changes to individual subroutines which have not been correlated. This program is not currently operational at IDA. Operating versions of the program exist at DCPACC and CCTC which are similar to this program but have more recent blast overpressure and INR distance algorithms.

```
      PROGRAM ANDANTE(INPUT,OUTPUT, PUNCH,TAPE0)
      COMMON/RUNVAL/NKEND,LABEL(50), JRAD,IPUNCH,NTYPE ,JPKTP
     1,LSTAPE,NT2,NT3,NT4,NT5,ADJSTF,T2,T3,T4,T5
C     RUNVAL USED IN INITOUT . NEWCTY,WPNOUT,CTYOUT   ,RADCALC,RUNPAR
      COMMON/WPNPR/MOD,DZER,SIG0,RMAX,PTG, YIELD,CEP,PSI,DEL,DDMAX,DMAX
     1,NOS    ,FMAXWH,MAXWP ,DESMAX
C     WPNPR USED IN WPNIN,FLPKHU,DZZ,WPNOUT ,ONEPASS ,OPTWPN,RADCAL
C     AND  TRDCL RADKL
      COMMON/PKARR/PK(110),DSQ(110), PKC(110),PSE(110),
     1DELDS(110),DELPS(110)
C     PKARR USED IN ONEPAS, OPTWPN, FLPKHU ,TRDCL     ,RADCAL
      COMMON/PAYAR/ PAYZ(100),XZ(100),YZ(100),PAYZT(100) ,IWP
C     PAYARR USED IN CITYOUT, OPTWPN ,PAYIN ,TRDCL      RADKL
      COMMON/CTYPAR/NAMEC(20),FLATC,FLONGC,FLFCT,TOTPOP,TPX,XSQ,YSQ,
     1SGTX,  SGTY,NTRCTS
C     CTYPARM USED IN NEWCTY,CTYOUT, OPTWPN ,PAYIN  ,ONEPASS,TRDCL
C     AND TRCTIN          ,RADKL
      COMMON/TRCTAR/ X(4000),Y(4000),POP(4000),V(4000)
C     TRCTAR USED IN NEWCTY,OPTWPN,ONEPASS ,TRDCL,TRCTINRNTRIN,RADKL
       COMMON/TRISP/XSP(200),YSP(200),NSP, POPSP(200),IVL(200)
      COMMON/INJAR/DSQI(110),PKCI(110) ,DSQK(110),PKCK(110) ,
     1DELPSI(110) ,DELPSI(110), DELDSK (110), DELPSK(110)
      COMMON/RADPAR/D50,SIG50,PF, C50,SIGC50,AR,BR,CR,PSINJ,SIGBC,SIGAL
      COMMON/IOAR/ MP,MQ,MS,MT,ITA,ITB,ITC

      MP = 5LINPUT
      MQ = 6LOUTPUT
      MS = 5LPUNCH
        MT = 5LTAPE0

      CALL RUNPAR
      IF( JRAD .NE. 1) GO TO 5
4     CONTINUE
      CALL RADCAL
      GO TO 1
5     CONTINUE
      CALL DG7SEL
      GO TO 1
1     CONTINUE
      STOP 6400
      END


      SUBROUTINE RUNPAR

C     READS VALUES ASSOCIATED WITH DEFINING THIS RUN

      COMMON/RUNVAL/NKEND,LABEL(50), JRAD,IPUNCH,NTYPE ,JPKTP
     1,LSTAPE,NT2,NT3,NT4,NT5,ADJSTF,T2,T3,T4,T5
      COMMON/IOAR/ MP,MQ,MS,MT,ITA,ITB,ITC
      READ(MP,1) (LABEL(I), I = 1,20)
1     FORMAT(20A4)
      READ(MP,2)JRAD
2     FORMAT(I10)
C     JRAD = 0 DG7CALC, EQUAL 1 DO SPECIAL RADIATION CALC
3     CONTINUE
      READ(MP,7)NKEND,IPUNCH
7     FORMAT( A4,6X,I10)
C     IPUNCH EQ 0 PUNCH OFF EQ 1 PUNCH ON

      READ( MP,8) JPKTP,ADJSTF
```

AD-A040 945   INSTITUTE FOR DEFENSE ANALYSES ARLINGTON VA   PROGRAM --ETC   F/G 15/4
DOCUMENTATION OF CURRENT IDA COMPUTER MATERIAL DEVELOPED FOR DC--ETC(U)
JAN 77   L A SCHMIDT   DCPA01-76-C-0213

UNCLASSIFIED   IDA/HQ-77-19225   NL

2 OF 3

AD
A040945

```
8       FORMAT(I10,F10.2)
C       IF JPKTP = 1 REGULAR OLD DGZSEL PK DIST CURVE, IF0 DIRECT PK
C       DISTANCE CALCULATION
        READ(MP,9)LSTAPE
9       FORMAT(I10)
C       LSTAPE =1 WRITE SUMMARY RESULTS ON TAPE 0 DONT

        RETURN
4       CONTINUE
        RETURN
        END


        SUBROUTINE DGZSEL

C       DOES THE REGULAR DGZSEL CALCULATION

        COMMON/RUNVAL/NNEND,LABEL(50), JRAD,IPUNCH,NTYPE ,JPKTP
       1,LSTAPE,NT2,NT3,NT4,NT5,T1,T2,T3,T4,T5

        CALL WPNIN
        CALL FLPKHU
        CALL INITOU
        CALL WPNOUT
        READ (MP,5) NMSH,NWOPT
5       FORMAT(2I10)
C       NMSH NO OF MESHES IN DGZSEL OPT  NWOPT 1          PUNCH AS OPT 0 DONT
C       NMESH = 6 USUALLY 12 FOR SMALL LETH RADIUS
10      CONTINUE
        CALL NEWCTY (IPASS )
        IF(IPASS .EQ. 1) GO TO 300
        CALL OPTWPN(NMSH,NWOPT)
        CALL CTYOUT
        GO TO 10
300     CONTINUE
        RETURN
        END


        SUBROUTINE RADCAL
        COMMON/RUNVAL/NNEND,LABEL(50), JRAD,IPUNCH,NTYPE ,JPKTP
       1,LSTAPE,NT2,NT3,NT4,NT5,T1,T2,T3,T4,T5
        COMMON/PKARR/PK(110),DS0(110), PKC(110),PSE(110),
       1DELDS(110),DELPS(110)
         COMMON/TRISP/XSP(200),YSP(200),NSP, POPSP(200),IVL(200)
        COMMON/WPNPR/MOD,DZER,SIGD,PMAX,PTG, YIELD,CEP,PSI,DEL,DDMAX,DMAX
       1,NDS    ,FMAXWP,MAXWP ,DESMX
        COMMON/RADPAR/DE0,STG50,PF, C50,SIGC50,AR,BR,CR,PSINJ,SIGBC,SIGAL
        COMMON/INJAR/DSCI(110),PKCI(110) ,DSQK(110),PKCK(110),
       1DELDSI(110) ,DELPSI(110), DELDSK (110), DELPSK(110)
        COMMON/IOAR/ MP,MQ,MS,MT,ITA,ITB,ITC

        CEPSV = 123456.789
        PSISV = 0.
        PSINJV = 0.
        YIELDV = 0.
        NTYPEV = 0
        LSTC = 15
29      CONTINUE
        READ(MP,25)LSTC
25      FORMAT (I10)
```

3

```
      IF (LSTC .NE. 3) GO TO 45
      CALL CHANGE
      GO TO 46
45    CONTINUE
C     IF LSTC = 0 READ IN NEW CITY AND WPN LOCS,1 DONT -1 EXIT
C     IF LSTC = -2 READ IN NEW WPN LOCS ONLY
C     IF LSTC EQUAL 3 USE CHANGE TO BRING IN A FEW NEW VARIABLES
      IF (LSTC  .EQ. -1) GO TO 26
       CALL WPNIN
      READ(MP,7)NSP
7     FORMAT(I10)
      READ(MP,4)D50,SIG50,PF, C50, SIGC50,PSINJ ,SIGBC,SIGBL
4     FORMAT( /,8F10.4)
      READ(MP,5) AR,BR,CR
5.    FORMAT(3F20.6)
46    CONTINUE
       CALL INITOU
       CALLWPNOUT
      WRITE(MO,8)
8     FORMAT(1H0,85H  D50      SIG OF D PROCT FAC  RAD CAS50 SIG OF CAS
     1BLST CAS  SG BL CAS SG BL LETH                      )
      WRITE(MO,4)D50,SIG50,PF,C50, SIGC50,PSINJ ,SIGBC,SIGBL
      IF((AR.NE.1.1).AND.(AR.NE.1.2) .AND.(AR.NE. 1.3)) GO TO 51
      WRITE(MO,52) AR
52    FORMAT(1H0, 59HSLANT RANGE INIT. RADIATION CALCULATION FOR AIR DEN
     1SITY OF       ,F4.1  )
      GO TO 53
51    CONTINUE
      WRITE(MO, 54) AR,BR,CR
54    FORMAT(1H0, 56HPRESSURE INIT. RADIATION CALCULATION FOR VALUES OF
     1AR = ,F12.5, 7H, BR =  ,F12.5,11H, AND CR =  ,F12.5   )
53    CONTINUE
      IF (LSTAPE .NE.1) GO TO 99
      WRITE(MT,4)D50,SIG50,PF,C50, SIGC50,PSINJ
      WRITE(MT,5) AR,BR,CR
99    CONTINUE
      IF( JPKTP .NE. 1) GO TO 31
       CALL FLPKOU
      WRITE(MO,68) PSI,DZER, PMAX
68    FORMAT( 43H VALUES OF LETHAL PSI DZER PMAX ARE               ,3F12.4)
      DO 12 I = 1,30
      PKCK(I) = PKC(I)
      DSNK(I) = DSN(I)
      DELPSK(I) = DELPS(I)
      DELPSK(I) = DELPS(I)
12    CONTINUE
      PSIST = PSI
      PSI = PSINJ
      CALL FLPKHJ
      WRITE(MO,69) PSI,DZER, PMAX
69    FORMAT(43H VALUES OF INJURY PSI DZER PMAX ARE                ,3F12.4)
      DO 13 I = 1,30
      PKCI(I) = PKC(I)
      DSNI(I) = DSN(I)
      DELPSI(I) = DELPS(I)
      DELPSI(I) = DELPS(I)
13    CONTINUE
      PSI = PSIST
      GO TO 39
31    CONTINUE
      IF ( YIELDV .NE. YIELD) GO TO 39
      IF( NTYPE .NE. NTYPEV) GO TO 38
      IF( CEPSV.NE.CEP) GO TO 38
      IF(PSISV .NE.PSI) GO TO 38
      IF( PSINJV .EQ. PSINJ) GO TO 39
38    CONTINUE
      CALL FLPKA
      YIELDV = YIELD
      NTYPEV = NTYPE
      CEPSV = CEP
      PSISV = PSI
      PSINJV = PSINJ
39    CONTINUE
      IF( LSTC .EQ. 0)    GO TO 27
      GO TO 28
27    CONTINUE
      CALL NEWCTY(IPASS)
      IF( IPASS .EQ. 1) GO TO 26
       CALL PAYIN
28    CONTINUE
      IF ( LSTC .EQ. -2) GO TO 41
      GO TO 42
41    CONTINUE
      CALL PAYIN
42    CONTINUE
      JKL = 0
      IF(JKL .EQ. 0) GO TO 6
      CALL RGTRIN
       CALL TPCTIN
6     CONTINUE
      CALL RNTRIN
      CALL TRPCL
      CALL  RADKL
      IF (LSTAPE .NE.1) GO TO 90
      ENDFILE MT
90    CONTINUE
      GO TO 29
26    CONTINUE
      IF (LSTAPE .NE.1) GO TO 91
      ENDFILE MT
      ENDFILE MT
91    CONTINUE
      RETURN
      END

      SUBROUTINE CHANGE(LSTC)

C     TO SIMPLIFY THE INPUT OF VARIABLES FOR A STRING OF RUNS

      COMMON/RUNVAL/NNEND,LABEL(50), JRAD,IPUNCH,NTYPE ,JPKTP
     1,LSTAPE,NT2,NT3,NT4,NT5,ADJSTF,T2,T3,T4,T5
      COMMON/WPNPR/MOD,DZER,SIGD,PMAX,PTG, YIELD,CEP,PSI,DEL,DDMAX,DMAX
     1,NDS    ,FMAXWP,MAXWP ,DESMX
       COMMON/TRTSP/XSP(200),YSP(200),NSP, POPSP(200),IVL(200)
      COMMON/RADPAR/DE0,SIG50,PF, C50,SIGC50,AR,BR,CR,PSINJ,SIGBC,SIGBL
      COMMON/TOAR/ MP,MQ,MS,MT,ITA,ITB,ITC
      INTEGER LSTC
1     CONTINUE
      READ( MP,10) VARB,IBLK,ILST
10    FORMAT(F 10.0,I2,I2 )
      GO TO (20,30,40,60,70,80 ,90                    ),IBLK
```

5

```
20      CONTINUE
C       TERMINATE CHANGES ON IBLK = 1
        RETURN

30      CONTINUE
C       WPNPR VARIABLES
        GO TO (31,32,33,34,35 ),ILST
31      YIELD = VARB
        GOTO1
32      CEP = VARB
        GOTO1
33      PSI = VARB
        GOTO1
34      DEL = VARB
        GOTO1
35      FMAXNP = VARB
        MAXNP = VARB
        GOTO1

40      CONTINUE
C       RADPAR VAZRIABLES
        GO TO (41,42,43,44,45,46,47,48,49,50,51 ),ILST
41      DEC = VARB
        GOTO1
42      SIG50 = VARB
        GOTO1
43      PF = VARB
        GOTO1
44      C50 = VARB
        GOTO1
45      SIGC50 = VARB
        GOTO1
46      AR = VARB
        GOTO1
47      BR = VARB
        GOTO1
48      CR = VARB
        GOTO1
49      PSINJ = VARB
        GOTO1
50      SIGHC = VARB
        GOTO1
51      SIGBL = VARB
        GOTO1

60      CONTINUE
C       RUNPAR VARIABLES
        GO TO (61,62,63,64,65,66,67),ILST
61      JRAD = VARB
        GOTO1
62      IPUNCH = VARB
        GOTO1
63      NTYPE = VARB
        GOTO1
64      JSKTP = VARB
        GOTO1
65      LSTAPE = VARB
        GO TO 1
66      ADJSTF = VARB
        GO TO 1
67      READ(NP,68)(LABEL(I),I = 1,20)
```

```
68      FORMAT(20A4)
        GOTO1

70      CONTINUE
C       TRCSP VARIABLES
        NSP = VARB
        GO TO 1

80      CONTINUE
C       SET LSTC
        LSTC = VARB
        GOTO1

90      CONTINUE
C       SPECIAL CASES
        GO TO  ( 91,92,93,97,98 ), ILST
C       TO READ ALL THREE ABC VARIABLES
91      READ( MP,94)AR,BR,CP
94      FORMAT(3F20.6)
        GOTO1
C       TO READ PSI AND PSINJ AND RETURN
92      CONTINUE
        READ(MP,95)PSI,PSINJ
95      FORMAT(2F10.0)
        RETURN
93      CONTINUE
C       TO USE PF VALUE AND RETURN
        PF = VARB
        RETURN
97      CONTINUE
C       READ PSI AND PSINJ AND THEN BLOCK 1 CARD
        READ(MP,95)PSI,PSINJ
        GOTO1
98      CONTINUE
C       SET PSI TO VALUE READ IN AND PSINJ TO 1/3 AND GO TO 1
        PSI = VARB
        PSINJ = VARB/3.
        GOTO1
        END
```

Nevins Compatible

```
        SUBROUTINE TRUCL

C       NEVINS COMPATIBLE
C       LAST REVISED NOV. 17, 1972

C       TO CALCULATE PSI THRM RAD ON A TRACT FROM PRELOCATED WEAPONS
C       ASSUME WEAPON LOCATIONS ARE IN NAUT. MILES ABOUT CITY CG

        COMMON/TMPAND/JPAD,NNEND,IPNCHA,IPUNCH,JPKTP,ADJSTF,LSTAPE,LSTC
       1 ,NSA,DFSMX,FMAXWP
        COMMON/WPNPRB/IWP,XZ(150),YZ(150),BLT(300)
        COMMON/ST44TA/X(4000),Y(4000),POP(4000),V(4000)
        COMMON/CITYPR/NAMEC(20),BLD(2), TOTPOP,BLE(30),NTRCTS
        COMMON/VULPR/BLAC(2),PSINJ,BLAH(5),D50,SIG50,C50,SIGC50,
       1 BLAD(46),NIWFL, PF, BLAF(19),BLAG(30)
        COMMON/WPNPR/ BLP(2),DEL,BLQ(7), NTYPA,BLR(13),YLDNA,BLS
        COMMON/PKPR/BLX, PKCK(30),DELPSK(30),DSQK(30),DELDSK(30),
       1 PKCI(30),DELPSI(30),DSQI(30),DELDSI(30) ,BLY
        COMMON/EFFCAL/BLI,YLDNU,BLJ,LONG,NTYPE,DIST,BLK,PSI,BLL,THRM,
       1THRPVS, RAD, BLM(13)
```

```
      COMMON /TRAN/RNC,RNW
      COMMON/TOPR/ALH,NO,ALHA(16)

      RNO = 4.6
      YLONU = YLONA
      NTYPE = NTYPA
C     ASSUME INFINITE VISIBILITY
      THRVS = 1.
      LONG = 1
      NSP = IABS(NSA)
      XNSP = NSP
      XNTRT = NTRCTS
      DEL = XNTRT/XNSP
      NDEL = DEL
      ITRN = -NDEL + 6
      DO  100 ICL = 1, NSP
      SUMTOT = 1.
      UNINJ = 1.
      PSIMX = 0.
      RADCUM = 0.
      RADMX = 0.
       THMX = 0.
       THCM = 0.
      PSYN = 0.
      IF (NSA .LT. 0) GO TO 32
      ITRN = ITRN + NDEL
      GO TO 33
32    CONTINUE
      CALL CALRN
      TRN = XNTRT*RNW
      ITRN = TRN
33    CONTINUE
      XPT = X(ITRN)
      YPT = Y(ITRN)
      POPSP = POP(ITRN)
      WRITE(NO,23)(NAMEC(I),I = 1,20),TOTPOP,NTRCTS
23    FORMAT( 1H1 ,/,43H INDIVIDUAL TRACT CALCULATIONS FOR CITY OF
     1,20A4,/,15H WITH POPN. OF    ,F10.0 ,  8H HAVING    , I5,
     2    7HTRACTS    ,//)
      WRITE(NO,18)ICL,XPT, YPT,POPSP, ITRN
18    FORMAT(1H0, 5H NO.  ,I4,3X, 7HEAST =   ,F8.4,3X, 6HNOR =  F8.4,
     1 3X, 6HPOP =  ,F8.1,3X,9HTRACT NO.    ,I5)
      WRITE(NO,25)
24    FORMAT( 1H0, 3HWPN,6H DIST , 9H   PSI   , 9H MAX PSI , 6H  PK  ,
     16H TOTPK, 6H PINJ  , 8H TOTPT  , 7H  RAD   ,7H MXRAD  ,7H CUMRAD ,
     26H PRADK  ,6H PRADI    ,
     3 7H  THRM,  7H MXTHRM, 7H CUMTHM,                        6H SYNRG )
      DO 110 IWPN = 1,IWP
      TX = XPT - XZ(IWPN)
      TY = YPT - YZ(IWPN)
      SDIST = TX*TX + TY*TY
      DIST = SQRT(SDIST)
      CALL PROMPT
      IF(PSI .GT. PSIMX) PSIMX = PSI
      DD = SDIST
      J = 1
164   CONTINUE
      J = J + 1
      IF(DSQK(J).GT. DD) GO TO 164
      P    = PKCN(J) + (DD - DSQK(J))*DELPSK(J)/DELDSK(J)
```

8

```
      PKw = P
      P = P*DEL
      J = ,
166   CONTINUF
      J = J + 1
      IF(DSQI(J).GT. DD) GO TO 166
      P    = PKCI(J) + (DD - DSQI(J))*DELPSI(J)/DELUSI(J)
      PIW = P
      SURTOT = SURIOT*(1. - PKW)
      PKTOT = 1. - SURTOT
      UNINJ = UNINJ*(1. - PIW)
      PITOT = 1. - UNINJ
      IF( RAD .GT. HACMX) RADMX = RAD
      RADCUM = RADCUM + RAD
      TVAL = RADCUM/PF
      TVL = (TVAL - D50)/SIG50
      PRAD = CUMNOR(TVL)
      TINJ = (TVAL - C50)/SIGC50
      PRADI = CUMNUR(TINJ)
      IF(THRM .GT.THMX) THMX = THRM
      THCM = THCM + THRM
      RADP = RAD
      IF(RAD .GT. 99999.) RADP = 99999.
      RADMXP = RADMX
      IF( RADMXP .GI. 99999.)  RADMXP = 99999.
      RADCUP = RADCUM
      IF( RADCUP .GT. 99999.) RADCUP = 99999.
      THRMP = THRM
      IF(THRM .GT. 99999.) THRMP = 99999.
      THMXP = THMX
      IF( THMX .GT. 99999.) THMXP = 99999.
      THCMP = THCM
      IF( THCM .GT. 99999.) THCMP = 99999.
      IF( PSIMX .GI. 1.25*PSINJ .AND.RADCUM.GT.(C50 + SIGC50)) PSYN =1.
      WRITE(MO,20)IWPN,DIST,PSI,PSIMX,PKW,PKTOT, PIW,PITOT, RADP,
     1RADMXP,RADCUP, PRAD,PRADI,THRMP,THMXP,THCMP,PSYN
20    FORMAT( 1H , I3, F6.2, 2F9.2, 4F6.3, 3F7.1,2F6.3,3F7.1,F6.3)
110   CONTINUF
100   CONTINUF
      RETURN
      END
```

Nevins Compatable

```
      SURROUTINE RADKL

C     NEVUNS COMPATABLE
C     LAST REVISED NOV. 16, 1972

C     COMPUTES AVERAGE KILL BY RADIATION
C     LETHAL AGENTS ARE BLAST AND THERNAL RADIATION

      DIMENSION RADAR(1000)   ,WELBAR(1000)
      DIMENSION PTAR(13,100)
      DIMENSION QAR(16,100), EA(15) , EAI( 15)
      COMMON/TMPAND/JRAD,NNEND,IPNCHA,IPUNCH,JPKTP,ADJSTF,LSTAPE,LSTC
     1 ,NSP,DFSMX,FMAYWP
      COMMON/WPNPRB/IWP,XZ(150),YZ(150),BLT(300)
      COMMON/ST44TA/X(4000),Y(4000),POP(4000),V(4000)
      COMMON/CITYPR/NAMEC(20),BLD(2), TOTPOP,BLE(30),NTRCTS
      COMMON/VULPP/ BLAC(8),D50,SIG50,C50,SIGC50, BLAD(46),NIPFL,PF,
     1 BLAF(19), BLAG(30)
```

```
      COMMON/WPNPP/ BLP(2),DEL,BLO(7), NTYPA,BLR(13),YLDNA,BLS
      COMMON/PKPR/BLX, PKCK(30),DELPSK(30),DSWK(30),DELDSK(30),
     1 PKCI(30),DELPSI(30),DSQI(30),DELDSI(30) ,BLY
      COMMON/EFFCAL/BLI,YLDNU,BLJ,LONG,NTYPE,DIST,BLK,PSI,BLL,THRM,
     1THRPVS, RAD, BLM(13)
      COMMON/TOPR/BLH, NO,BLAA(7),MT,BLAB(8)

      YLDNU = YLDNA
      NTYPE = NTYPA
C      ASSUME INFINITE VISIBILITY TO GET MAX THERMAL DOSEAGE
      THRPVS = 1.
      LONG = 1
      PSYNRG = 1.
      PCORL = .5
      WRITE(NO,3)
3     FORMAT(1H1,//,50X, 16HTOTAL CITY EFFECTS    )
      WRITE(NO,4) (NAMEC(I),I = 1,20)
4     FORMAT(1H0, 20X, 13HCITY NAME IS    ,20A4)
      WRITE(NO,35)
35    FORMAT(1H0)
      WRITE(NO,30)
30    FORMAT( 1H ,16X,10HFATALITIES  13X,16X, 8HINJURIES
     1 17X,  18H  COMP FATALITIES     ,18H SYNERGISTIC FATL      )
      WRITE(NO,31)
31    FORMAT( 1H , 8X,9HRADIATION ,4X, 7X, 5HBLAST, 6X  ,
     1              8X,9HRADIATION ,4X, 7X, 5HBLAST, 8X ,
     2   18H  RAD. AND BLAST    ,18H  RAD. AND BLAST          )
      WRITE(NO,32)
32    FORMAT( 1H  ,3X, 18H LST WPN    TOTAL       ,18H   TOTAL    LST WPN
     1              ,3X, 18H LST WPN    TOTAL       ,18H   TOTAL    LST WPN
     2    ,3X      ,     18H LST WPN    TOTAL       ,18H  LST WPN   TOTAL  )
      IF (LSTAPE .NE.1) GO TO 90
      WRITE(MT,3)
      WRITE(MT,35)
      WRITE(MT,30)
      WRITE(MT,31)
      WRITE(MT,32)
90    CONTINUE
      DO 5 I = 1,NTARCTS
      V(I) = POP(I)
      DELBAR(I) = POP(I)
5     RADAR(I) = 0.
      RRAYL = 0.
      RIL = 0.
      BKRI =  .
      BKRSL = 0.
      BKTOT =0.
      RITOT = 0.
      BKRTOT = 0.
      BKRSTO = 0.
      DO 10  II = 1,IWP
      RRAY = 0.
      RIRAY = 0.
      BRAY   0.
      RIRAY = 0.
      BKRRAY = 0.
      BKRSRY =0.
      WKRKB    = 0.
      RIRKB = 0.
      BRRKB =     0.
      QKRIB = 0.
      QIRIB = 0.
      QURIB =   0.
      QKRUB = 0.
      QIRUB = 0.
      QURUB = 0.
      SMCOR = 0.
      DO 12 JJ = 1,NTRCTS
      TX = XZ(II) - X(JJ)
      TY = YZ(II) - Y(JJ)
      DISTS = TX*TX + TY*TY
      DIST = SQRT(DISTS)
      CALL PROMPT
      RADAR(JJ) = RADAR(JJ) + RAD
      TT = RADAR(JJ)*DEL/PF
      TTT = (TT - D50)/SIG50
      PKR= CUMNOR(TTT)
      RPAY = POP(JJ) * PKR+ RPAY
      TTI = ( TT - V50)/SIGC50
      PIR = CUMNOR(TTT)
      RIPAY = POP(JJ)*PIR + RIPAY
      PIIR = PIR - PKR
      PUR = 1. - PIR
      DD = DISTS
C     TO SPEED UP CALC - NEEDS PKCK(2) = 0.
      IF( DSQK(2) .GT.DD) GO TO 162
      PKB = 0.
      GO TO 163
162   CONTINUE
      J = 2
164   CONTINUE
      J = J + 1
      IF(DSQK(J).GT. DD) GO TO 164
      P    = PKCK(J) + (DD - DSQK(J))*DELPSK(J)/DELDSK(J)
      PKB = P*DEL
163   CONTINUE
      IF( DSQI(2) .GT. DD) GO TO 168
      PIR = 0.
      GO TO 169
168   CONTINUE
      J = 2
166   CONTINUE
      J = J + 1
      IF(DSQI(J).GT. DD) GO TO 166
      P    = PKCI(J) + (DD - DSQI(J))*DELPSI(J)/DELDSI(J)
      PIR = P*DEL
169   CONTINUE
      BPAY = BPAY + V(JJ)*PKB
      BIPAY = BIPAY + WELBAR(JJ)*PIR
C     REDUCE  NUMBER OF BLAST SURVIVORS AND UNINJURED
      V(JJ) = V(JJ) *(1. - PKB)
      WELBAR(JJ) = WELBAR(JJ)*(1. - PIB)
      BKRPAY = BKRPAY + V(JJ)*PKR
      BINJO =  V(JJ) - WELBAR(JJ)
C     ASSUME THAT IF INJURED BY BOTH BLAST AND RADIATION THIS IS FATAL
C     FOR SYNERGISTIC EFFECTS CALCULATION. OTHER DATA CAN BE USED HERE
C     IF DESIRED
      BKRSPY = BKRSPY + BINJO*  PIIR      *PSYNRG
      BKL = POP(JJ) - V(JJ)
      QKRKB = QKRKB + BKL*PKR
      QIRKB = QIRKB + BKL*PIIR
```

11

```
      QURKR =  QURKR + BKL*PUR
      QKRIR = QKRIR + BINJO*PKR
      QIRIR = QIRIR + BINJO*PIIR
      QURIR =  QURIR + BINJO*PUR
      QKRUR = QKRUR + WELBAR(JJ) *PKR
      QIRUR =  QIRUR + WELBAR(JJ)*PIIR
      QURUB = QURUB + WELBAR(JJ) *PUR
12    CONTINUE
      DRPAY = RPAY - RPAYL
      RI = RIPAY - RPAY
      DRI = RI - RIL
      BKTOT = BKTOT + BPAY
      BINPN = BIPAY - BPAY
      BITOT = BITOT + BINPN
      BKRTOT = BKTOT + BKRPAY
      DBKRT = BKRTOT - BKRL
      BKRSTO = BKRTOT + BKRSPY
      DBKRST = BKRSTO - BKRSL
      IF(DRPAY.LE.BPAY) GO TO 160
      FATCO = BPAY + (DRPAY-BPAY)*(1.- PCURL)
      SMCOR = SMCOR + FATCO
160   CONTINUE
      WRITE(MO,48) II,DRPAY,RPAY,BKTOT,BPAY,DRI,RI,BITOT,BINPN,DBKRT,
     ,BKRTOT,DBKRST,BKRSTO
48    FORMAT( 1H ,I3, 4F9.0,2X,4F9.0,2X,2F9.0,2X,2F9.0)
      IF (LSTAPE .NE.1) GO TO 91
      WRITE(MT,48) II,DRPAY,RPAY,BKTOT,BPAY,DRI,RI,BITOT,BINPN,DBKRT,
     ,BKRTOT,DBKRST,BKRSTO
91    CONTINUE
      PTAR(1,II) = RIPAY
      PTAR(2,II) = BITOT + BKTOT
      PTAR(3,II) = TOTPOP - QURUB
      PTAR(4,II) = QIRIR
      PTAR(5,II) = RPAY - BKRPAY
      PTAR(6,II) = QURUB
      PTAR(7,II)= 100.*PTAR(6,II)/TOTPOP
      PTAR(8,II) = TOTPOP - BKRTOT
      PTAR(9,II)= 100.*PTAR(8,II) /TOTPOP
      PTAR( 10,II) = TOTPOP - BKTOT
      PTAR(11,II) =  100.*PTAR(10,II) /TOTPOP
      PTAR(12,II) = TOTPOP - RPAY
      PTAR(13,II) = 100.*PTAR(12,II)/TOTPOP
      QAR( 1,II)=  RPAY
      QAR( 2,II)=  RI
      QAR( 3,II)=  TOTPOP - RPAY - RI
      QAR( 4,II)=  BKTOT
      QAR( 5,II)=  BITOT
      QAR( 6,II)=  TOTPOP - BKTOT - BITOT
      QAR( 7,II)=  QKRKR
      QAR( 8,II)=  QKRIR
      QAR( 9,II)=  QKRUR
      QAR(10,II)=  QIRKR
      QAR(11,II)=  QIRIR
      QAR(12,II)=  QIRUR
      QAR(13,II)=  QURKR
      QAR(14,II)=  QURIR
      QAR(15,II)=  QURUR
      QAR(16,II) = SMCOR
      RPAYL = RPAY
      RIL = RI
      BKRL = BKRTOT
```

1✓

```
      BKRSL = BKRSIO
10    CONTINUF
      WRITE(MO,52)
52    FORMAT(1H1,//,50X,25H MORE TOTAL CITY EFFECTS            )
      WRITE(MO,54)
54    FORMAT(1H0,11X,10HCASUALTIES  ,13X,9HINJURIFS       ,10HFATALITIES
     1   ,7X,9HUNINJURED,7X ,
     2 22H  *    *    *    *            ,9HSURVIVORS ,
     3  21H  *    *    *    *            )
      WRITE (MQ,56)
56    FORMAT( 1H  ,5X, 4HRAD.  ,  5X, 5HBLAST  ,4X, 5HCOMB.  ,6X,
     1 9HBOTH R B  ,9HBOTH R R  ,10X, 8HCOMBINED, 12X,8HCOMBINED,  11X,
     2 5HBLAST, 11X, 9HRADIATION   )
      WRITE (MQ,58)
58    FORMAT( 1H , 22X, 8H(R OR B) ,4X,
     1 9H(R AND B)  ,9H(R AND B)  , 6X, 5HTOTAL, 3X, 7HPERCENT  ,
     2 5X,5HTOTAL ,3X, 7HPERCENT ,3X, 5HTOTAL, 3X,7HPERCENT,3X,
     3  5HTOTAL, 3X,7HPERCENT  )
      IF (LSTAPE .NE.1) GO TO 92
      WRITE(MT, 52)
      WRITE(MT, 54)
      WRITE(MT, 56)
      WRITE(MT, 58)
92    CONTINUF
      DO 61 IJ = 1,IWP
      WRITE(MO, 62) IJ,PTAR(1,IJ), PTAR(2,IJ) , PTAR(3,IJ), PTAR(4,IJ),
     1PTAR(5,IJ)  ,PTAR(6,IJ) ,  PTAR(7,IJ),  PTAR(8,IJ),    PTAR(9,IJ),
     2PTAR(10,IJ) , PTAR(11,IJ) , PTAR(12,IJ), PTAR(13,IJ)
62    FORMAT( 1H  ,I3, 3F9.0, 4X, 2F9.0,4X,F9.0, F7.3,4X,F9.0,
     1F7.3     ,2X, F9.0,F7.3, 2X, F9.0,F7.3    )
      IF (LSTAPE .NE.1) GO TO 93
      WRITE(MT, 62) IJ,PTAR(1,IJ), PTAR(2,IJ) , PTAR(3,IJ), PTAR(4,IJ),
     1PTAR(5,IJ)  ,PTAR(6,IJ) ,  PTAR(7,IJ),  PTAR(8,IJ),    PTAR(9,IJ),
     2PTAR(10,IJ) , PTAR(11,IJ) , PTAR(12,IJ), PTAR(13,IJ)
93    CONTINUF
61    CONTINUF
      WRITE( MQ,79)
79    FORMAT( 1H1,  //,  50X, 29HSTILL MORE TOTAL CITY EFFECTS        )
      WRITE(MO, 75)
75    FORMAT(1H  ,30X,55HSINGLE EFFECTS  (A B)  ARE PERCENT AND 1 - PERC
     1ENT              )
      WRITE (MQ,74)
74    FORMAT( 1H0 , 11X, 15H  *    *    *    , 9HRADIATION
     1,  15H  *    *    *    ,10X, 17H   *    *    *        ,5HBLAST,
     217H  *    *    *         ,10X,10HCORRELATED        )
      WRITE (MQ,73)
73    FORMAT( 1H , 5X,6X, 10HFATALITIES  ,6X, 8HINJURIES, 6X,
     19HUNINJURED ,11X,    10HFATALITIES, 6X,8HINJURIES    ,6X,
     2 9HUNINJURED, 10X, 10HFATALITIES     )
      IF (LSTAPE .NE.1) GO TO 94
      WRITE(MT, 79)
      WRITE(MT, 75)
      WRITE(MT, 74)
      WRITE(MT, 73)
94    CONTINUF
      DO 71 K = 1,IWP
      WRITE(MO,72) K,(QAR(I,K),I = 1,6), QAR(16,K)
72    FORMAT(1H0 ,I3,2X,3F15.0,5X,3F15.0,10X,F10.0 )
      IF (LSTAPE .NE.1) GO TO 95
      WRITE(MT,72) K,(QAR(I,K),I = 1,6), QAR(16,K)
95    CONTINUF
```

```
      DO 77 I = 1,6
      EA(I) = 100.*QAR(I,K)/TOTPOP
      EAI(I) = 100. - EA(I)
77    CONTINUE
      WRITE(MO,76)(EA(I),EAI(I),I = 1,6)
76    FORMAT( 1H  ,11X, 2F5.1, 5X,2F5.1,5X,  2F5.1,  9X, 2F5.1,5X,
     1 2F5.1, 5X, 2F5.1       )
      IF (LSTAPE .NE.1) GO TO 96
      WRITE(MT,76)(EA(I),EAI(I),I = 1,6)
96    CONTINUE
71    CONTINUE
      WRITE(MO,85)
85    FORMAT(1H1, //, 50X,32HAND EVEN MORE TOTAL CITY EFFECTS       )
      WRITE(MO,86)
86    FORMAT( 1H  ,25X, 70HCOMBINED EFFECTS.  NUMBERS HAVING INDICATED T
     1YPE OF BOTH EFFECTS                    )
      WRITE( MO,83)
83    FORMAT( 1H0,  16HRADIATION EFFECT  ,17X, 5HFATAL ,32X,7HINJURED,
     1 30X, 9HUNINJURED      )
      WRITE( MO,84)
84    FORMAT( 1H0, 16H    BLAST EFFECT,5X,   5HFATAL, 7X, 7HINJURED,
     13X,9HUNINJURED   ,7X, 5HFATAL, 7X, 7HINJURED, 3X, 9HUNINJURED,
     27X,  5HFATAL ,7X,  7HINJURED, 3X,9HUNINJURED      )
      IF (LSTAPE .NE.1) GO TO 97
      WRITE (MT, 85 )
      WRITE (MT, 86 )
      WRITE (MT, 83 )
      WRITE (MT, 84 )
97    CONTINUE
      DO 81 K = 1,NP
      WRITE(MO,82) K,( QAR(I,K) ,I = 7,15)
82    FORMAT( 1H0,1X,9HWPN NO. (  ,I3,1H) ,2X,3F12.0,2X,3F12.0,
     12X, 3F12.0)
      DO 87 I = 7,15
      EA(I) = 100.*QAR(I,K)/TOTPOP
      EAI(I) = 100. - EA(I)
87    CONTINUE
      WRITE(MO,88) (EA(I),EAI(I),I = 7,15)
88    FORMAT(1H ,16X,    3X,2F5.1,2X,2F5.1,2X,2F5.1,1X,
     1                   3X,2F5.1,2X,2F5.1,2X,2F5.1,1X,
     2                   3X,2F5.1,2X,2F5.1,2X,2F5.1     )
      IF (LSTAPE .NE.1) GO TO 98
      WRITE(MT,82) K,( QAR(I,K) ,I = 7,15)
      WRITE(MT,88) (EA(I),EAI(I),I = 7,15)
98    CONTINUE
81    CONTINUE
      RETURN
      END


      SUBROUTINE WPNIN

C     INPUTS VALUES ASSOCIATED WITH THE WEAPONS

      COMMON/WPNPF/MOD,DZER,SIGD,PMAX,PTG, YIELD,CEP,PSI,DEL,DDMAX,DMAX
     1,NWS   ,FMAXWP,MAXWP ,DESMX
      COMMON/IOAR/ MP,MQ,MS,MT,ITA,ITB,ITC
      COMMON/PUNVAL/NEND,LABEL(20), JRAD,IPUNCH,NTYPE ,JPKTP
     1,LSTAPE,NT2,NT3,NT4,NT5,T1,T2,T3,T4,T5
      READ(MP,2)(YIELD,CEP,PSI,DEL,DESMX,FMAXWP
2     FORMAT( /, 4F10.5,F10.0,F10.5)
```

```
      MAXWP=FMAXWP
      IF ( PST .LT. 0.) GO TO 5
      NTYPE = -1
      GO TO 4
5     NTYPE = 1
4     CONTINUF
      IF( CEP .GE. 0.) GO TO 6
      CEP = -CEP
      NTYPE = 0
6     CONTINUF
      RETURN
      END

      SUBROUTINE NEWCTY  (IPASS)

C     NEVUNS STANDARD
C     LAST REVISED NOV. 9, 1972

C     READS TRACT DATA FROM INPUT MEDIUM MA WHERE DATA IS IN SHORT
C     FORM WITH ONLY LAT,LONG,POPN FOR EACH TRACT. ASSUMES HEADER CARD
C     HAS CITY NAME AND A NEGATIVE LATITUDE IS AFTER THE LAST TRACT.
C     THE SUBROUTINE ALSO COMPUTES CENTER OF GRAVITY AND NS EW SIGMAS

      COMMON/CITYPR/NAMEC(20),BLA(2),TOTPOP,BLB,FLATC,FLONC,FLFCT,
     1BLC(3),SGTX,SGTY,BLD(21),NTRCTS
      COMMON/ST44TA/X(4000),Y(4000),POP(4000),V(4000)
      COMMON/TMPAND/JRAD,NNEND,IPNCHA,IPUNCH,JPKTP,ADJSTF,LSTAPE,LSTC
     1 ,NSP,DFSMX,FMAXWP
      COMMON/TOPR/BLE,MQ,MP,BLG(14)

      XSQ = 0.
      YSQ=0.
      FLATC = 0.
      FLONGC = 0.
      TOTPOP = 0.
      READ(MP,3) (NAMEC(I),I = 1,20)
3     FORMAT(20A4)
C     IFITHIS IS THE LAST CITY AND CITY NAME IS NNEND RETURN WITH
C     IPASS = 1, OTHERWISE IPASS = 0
      IPASS = 0
      IF( NAMEC(1) - NNEND)803,300,803
300   CONTINUF
      IPASS = 1
      RETURN
  803 CONTINUF
      I = 1

11    CONTINUF
      READ (MP,4)  Y(I),X(I), POP(I)
4     FORMAT( 2F10.3,F10.0)
C     TERMINATE CITY READ BY NEGATIVE LATITUDE
      IF(Y(I)) 6,5,5
5     CONTINUE
      FLATC = FLATC + Y(I)*POP(I)
      FLONGC = FLONGC + X(I) * POP(I)
      TOTPOP = TOTPOP + POP(I)
      I = I + 1
      GO TO 11

C     READING COMPLETED NOW COMPUTE STATISTICS AND FILL ARRAYS
```

NTYPE = -1 optimum airburst (lower of [illegible])
0 10 psi optimum airburst
1 Surface burst

15

```
6       CONTINUE
        NTRCTS = I - 1
        FLATC = FLATC/TOTPOP
        FLONGC = FLONGC/TOTPOP
        FLECT=COS (FLATC*3.14159/180.)
        DO 150 I=1,NTRCTS
        V(I)=POP(I)
        X(I)=(X(I)-FLONGC)*60.*FLECT
        Y(I)=(Y(I)-FLATC)*60.
        XSQ=XSQ+X(I)*X(I)*POP(I)
        YSQ=YSQ+Y(I)*Y(I)*POP(I)
  150   CONTINUE
        XSQ = XSQ /TOTPOP
        YSQ = YSQ/TOTPOP
        SGTX=SQRT (XSQ)
        SGTY=SQRT (YSQ)
```

Find population centroid and Standard Deviation

```
C       OUTPUT TRACT STATISTICS ON STANDARD OUTPUT MEDIUM
        WRITE(MO,27)
27      FORMAT( 1H1, ///////)
        WRITE(MO,21)(NAMEC(I),I = 1,15)
21      FORMAT(1H0, 34HNEW TRACT DATA FOR THE CITY NAMED     ,15A4)
        WRITE(MO,22) FLATC,FLONGC,TOTPOP,NTRCTS
22      FORMAT(1H  ,12HLATITUDE IS ,F8.5,4X,13HLONGITUDE IS   , F9.5, ,4X,
       1  20HTOTAL POPULATION IS ,F9.0,4X,13HNO. TRACTS IS   ,I5)
        WRITE (MO,23) SGTX,SGTY
23      FORMAT( 1H  ,32HSTD. DEV. IN E - W DIRECTION IS ,F9.5, 4X,
       132HSTD. DEV. IN N -S DIRECTION IS   ,F9.5 )
        RETURN
        END


        SUBROUTINE PAYIN

C       USED TO REFILL PAYARR FROM A PREVIOUS CALCULATION

        COMMON/PAYAR/ PAYZ(100),XZ(100),YZ(100),PAYZT(100) ,IWP
        COMMON/CTYPAR/NAMEC(20),FLATC,FLONGC,FLECT,TOTPOP,TPX,XSQ,YSQ,
       1SGTX,  SGTY,NTRCTS
        COMMON/IOAR/ MP,MO,MS,MT,ITA,ITB,ITC

        READ( MP,202) IWP
202     FORMAT(55X,I5)
        DO 215 K = 1,IWP
        READ(MP,203) J,PAYZT(J), PAYZ(J), XZ(J), YZ(J)
203     FORMAT( I5,4F15.5)
215     CONTINUE
        FLECT=COS (FLATC*3.14159/180.)
        DO 216 I = 1,IWP
        XZ(I) = (XZ(I) - FLONGC)* 60.*FLECT
        YZ(I) = (YZ(I) - FLATC)* 60.
216     CONTINUE
        RETURN
        END
```

For use with Blast-INR evaluation when no simultaneous optimization is desired

```
        SUBROUTINE RNTRIN

C       RANDOMLY SELECTS TRACTS FOR SPECIAL ARRAY

        COMMON/TRCTAR/ X(4000),Y(4000),POP(4000),V(4000)
```

For use with TROCL

```
         COMMON/TRTSP/XSP(200),YSP(200),NSP, POPSP(200),IVL(200)
        RNO = 4.6
        CALL CALRN(RNO,RNW)
        DO 10 J = 1,NSP
15      CONTINUE
        RNO = 0.
        CALL CALRN(RNO,RNW)
        TRN = 745.*RNW
        ITRN = TRN
        DO 12 K = 1,NSP
        IF( IVL(K) .EQ. ITRN) GO TO 15
12      CONTINUF
        XSP(J) = X(ITRN)
        YSP(J) = Y(ITRN)
        IVL(J) = ITRN
        POPSP(J) = POP(ITRN)
10      CONTINUE
        RETURN
        END

        SUBROUTINE RGTRIN
C       SELECTS TRACTS AT REGULAR INTERVALS FOR INPUT

        COMMON/TRCTAR/ X(4000),Y(4000),POP(4000),V(4000)
         COMMON/TRISP/XSP(200),YSP(200),NSP, POPSP(200),IVL(200)
        XNSP = NSP
        DEL = 745./XNSP
        NDEL = DEL
        ITRN = -NDEL + 6
        DO 20 J = 1,NSP
         ITRN = ITRN  + NDEL
        XSP(J) = X(ITRN)
        YSP(J) = Y(ITRN)
        IVL(J) = ITRN
        POPSP(J) = POP(ITRN)
20      CONTINUE
        RETURN
        END

         SUBROUTINE TRCTIN
C       TO READ INPUT VALUES OF TRACTS FOR SPECIAL ANALYSIS

         COMMON/TRISP/XSP(200),YSP(200),NSP, POPSP(200),IVL(200)
        COMMON/CTYPAR/NAMEC(20),FLATC,FLONGC,FLFCT,TOTPOP,TPX,XSQ,YSQ,
       ,SGTX,  SGTY,NTRCTS
        COMMON/TOAR/ MP,MQ,MS,MT,ITA,ITB,ITC
        I = 1
11      CONTINUF
        READ(MP,4)  YSP(I), XSP(I), POPSP(I)
4       FORMAT( 2F10.3,F10.0)
C       TERMINATE CITY READ BY NEGATIVE LATITUDE
        IF(YSP(I)) 6,5,5
5       CONTINUF
        I = I + 1
        GO TO 11
6       CONTINUF
        NSP = I - 1
```

For use with TROCL

For use with TROCL

```
      DO 150 I = 1,NSP
      XSP(I) = (XSP(I) - FLONGC) * 60.*FLFCT
      YSP(I) = ( YSP(I) - FLATC)*60.
150   CONTINUE
      RETURN
      END

      SUBROUTINE INITOU

C     INTITIAL PRINTOUT

      DIMENSION IOA(120)
      COMMON/RUNVAL/NKEND,LABEL(50), JRAD,IPUNCH,NTYPE ,JPKTP
     1,LSTAPE,NT2,NT3,NT4,NT5,ADJSTF,T2,T3,T4,T5
      COMMON/IOAR/ MR,MQ,MS,MT,ITA,ITB,ITC

      CALL TIME(CLTIM)
      CLTIM = ABS(CLTIM)
      RNO = CLTIM * 1.2
      CALL CALRN(RNO,RNW)
      WRITE(MO,18)
18    FORMAT(1H1)
      IF(IPUNCH .EQ. 0) GO TO 10
      WRITE(MS,1) (LABEL(I) , I = 1,20)
1     FORMAT(20A4)
10    CONTINUE
      RNO = 0.
      CALL CALRN(RNO,RNW)
      YLO = 17.+ 10.*RNW
      JLO = YLO
      ALO = 50./((40.-YLO)*(40.-YLO))
      RNO = 0.
      CALL CALRN(RNO,RNW)
      YM = 25. + 10.*RNW
      JM = YM
      RNO = 0.
      CALL CALRN(RNO,RNW)
      YTP = 15.*RNW
      JTP = YTP
      ATP = 60./((YM-YTP)*(YM-YTP))
      DO 21 JRN =JTP,40
      YRN = JRN
      IF(JRN .NE. 40) GO TO 20
      WRITE(MO,19)
19    FORMAT(1H+, 57X, 15HNEW ADDANTE RUN   )
20    CONTINUE
      IF( JRN .LT. JM) GO TO 62
      XT1 = 0.
      GO TO 63
62    CONTINUE
      XT1 = ATP*(YM - YRN)*(YM -YRN)
63    CONTINUE
      KT1 = XT1
      KT2 = 120- KT1
      XT2 = KT2
      XL1= ALO*(40.-YRN)*(40.-YRN)
      KL1 = XL1
      KL2 = 120- KL1
      XL2 =KL2
      DLX = 0.5*(XL1-XT1) +.00001
```

18

```
      MDX1 = 0.5*(AL1.XT1)
      MDX2 = 0.5*(AL2+ XT2)
      DO 22 KRN = 1,120
      XRN = KRN
      IF( KRN .GE. KT1) GO TO 23
      IOA(KRN) = 1L
      GO TO 22
23    CONTINUF
      IF( JRN .GT. JLC) GO TO 27
      IF( KRN .LE. KT2) GO TO 65
      IOA(KRL) = 1L
      GO TO 22
65    CONTINUF
      IF(KRN .GT. 60) GO TO 66
      XOFF = ABS(XRN-MDX1)
      GO TO 30
66    CONTINUF
      XOFF = ABS(XRN-MDX2)
      GO TO 30
27    CONTINUF
      IF(KRN .GT. KL1) GO TO 24
      XOFF = ABS(XRN - MDX1)
      GO TO 30
24    CONTINUF
      IF( KRN .GE. KL2) GO TO 25
      IOA(KRN) = 1L
      GO TO 22
25    CONTINUF
      IF( KRN .GT. KT2 ) GO TO 26
      XOFF = ABS(XRN - MDX2)
      GO TO 30
26    CONTINUF
      IOA(KRN) = 1L
      GO TO 22
30    CONTINUE
      RNO = 0.
      CALL CAIRN(RNO,RNW)
      XVL = 1. -((XOFF-DLX)/DLX)*((XOFF-DLX)/DLX) + 0.2*RNW
      IF(XVL .LE. 1) GO TO 31
      IOA(KRN) = 1L*
      GO TO 22
31    CONTINUF
      IF (XVL .LE. 0.9) GO TO 32
      IOA(KRN) = 1LX
      GO TO 22
32    CONTINUF
      IF(XVL .LE. 0.7) GO TO 33
      IOA(KRN) = 1LO
      GO TO 22
33    CONTINUF
      IF( XVL .LE. 0.5) GO TO 34
      IOA(KRN) = 1L+
      GO TO 22
34    CONTINUF
      IF(XVL.LE. 0.2) GO TO 35
      IOA(KRN)  = 1L.
      GO TO 22
35    CONTINUF
      IOA(KRN) =1L
22    CONTINUE
      WRITE(MO,38)(IOA(I),I = 1,120)
```

```
38      FORMAT( 1H ,5X,120A1)
21      CONTINUE
        WRITE(MO,11)(LABEL(I), I = 1,20)
11      FORMAT( 1H0,25X,20A4)
        WRITE(MO,41)
41      FORMAT(///)
        IF(JPKTP .NE. 1) GO TO 45
        WRITE(MO,43)
43      FORMAT(1H ,48X, 34HCALCULATE OPTIMIZED WEAPON LAYDOWN     )
        GO TO 46
45      CONTINUE
        IF(JPKTP .NE. 0) GO TO 46
        WRITE(MO,44)
44      FORMAT( 1H ,27X, 66HASSESS RESULTS OF BLAST AND RADIATION FROM A G
       1IVEN WEAPON LAYDOWN       )
46      CONTINUE
        WRITE(MO,41)
        RNO = 0.
        CALL CALRN(RNO,RNW)
        DO 51 JRN = 1,6
        YRN = JRN
        YVL = RNW*0.2+ 1.-SQRT(YRN/6.)
        DO 52 KRN = 1,120
        XRN = KRN
        RNO = 0.
        CALL CALRN(RNO,RNW)
        XVL =-ABS(60.-XRN)/120. + 0.5*RNW  +0.5
        VAL = XVL + YVL
        IF(VAL .LE. 1.7) GO TO 53
        IOA(KRN) = 1H*
        GO TO 52
53      CONTINUE
        IF( VAL .LE. 1.5) GO TO 54
        IOA(KRN) = 1HX
        GO TO 52
54      CONTINUE
        IF(  VAL .LE. 1.2) GO TO 55
        IOA(KRN) = 1HO
        GO TO 52
55      CONTINUE
        IF( VAL .LE. 1.0) GO TO 56
        IOA(KRN) = 1H+
        GO TO 52
56      CONTINUE
        IF( VAL .LE. 0.6) GO TO 57
        IOA(KRN) = 1H.
        GO TO 52
57      CONTINUE
        IOA(KRN) = 1H
52      CONTINUE
        WRITE(MO,38)(IOA(I), I = 1,120)
51      CONTINUE
        WRITE(MO,18)
        WRITE(MO,71)
71      FORMAT(////////)
        IF(JPKTP .NE.1) GO TO 72
        WRITE(MO,73)
73      FORMAT(1H ,95H   FOR THIS CALCULATION THE FITTED PK DISTANCE CURVE
       1 AS IN THE OLD PBZSEL CALCULATION IS USED               )
        GO TO 75
72      CONTINUE
```

20

```
      WRITE(MQ,74)ADJSTF
74    FORMAT( 1H ,79HDIRECT PK DISTANCE CALCULATION USING PK VS. PRESSUR
     1E AND PRESSURE VS. DISTANCE    ,/,27H NORMALIZING ADJUSTMENT IS
     2 ,  F6.4   ,42H OF DIFFERENCE FROM AREA OF LETHAL RADIUS      )
75    CONTINUE
      IF (IPUNCH .NE. 1) GO TO 76
      WRITE (MQ,77)
77    FORMAT( 1H0, 42HTHERE IS PUNCH CARD OUTPUT WITH THIS RUN      )
      GO TO 78
76    CONTINUE
      WRITE(MQ,79)
79    FORMAT(1H0,46HTHERE IS NO PUNCHED CARD OUTPUT WITH THIS RUN     )
78    CONTINUE
      IF (LSTAPE .NE. 1) GO TO 81
      WRITE(MQ,82)
82    FORMAT ( 1H0,78HA MAGNETIC TAPE OF SOME OUTPUT IS WRITTEN TO PRESE
     1RVE IT IN MACHINEABLE FORM         )
81    CONTINUE
      WRITE(MQ,84) NNEND
84    FORMAT( 1H0,65HTHE SYMBOLS USED AS A CITY NAME TO FLAG THE END OF
     1THE CITIES IS       ,A4)
      RETURN
      END


      SUBROUTINE WPNOUT

C     OUTPUTS  VALUES OF WEAPON PARAMETERS

      COMMON/RUNVAL/NNEND,LABEL(50), JRAD,IPUNCH,NTYPE ,JPKTP
     1,LSTAPE,NT2,NT3,NT4,NT5,T1,T2,T3,T4,T5
      COMMON/WPNPR/MOD,DZER,SIGD,PMAX,PTG, YIELD,CEP,PSI,DEL,DDMAX,DMAX
     1,NDS    ,FMAXWP,MAXWP ,DESMX
      COMMON/IOAR/ MP,MQ,MS,MT,ITA,ITB,ITC
      WRITE(MQ,5)
5     FORMAT(1H0,10HVALUES OF         ,60H YIELD        CEP          PSI
     1DEL      MAX RET    MAX WPNS             )
      WRITE(MQ,12)YIELD,CEP,PSI,DEL,DESMX,FMAXWP
12    FORMAT( 1H , 9X,4F10.5,F10.0,F10.5)
      IF ( IPUNCH .EQ. 0) GO TO 10
      WRITE(MS,2)               YIELD,CEP,PSI,DEL,DESMX,FMAXWP
2     FORMAT( 4F10.5,F10.0,F10.5)
10    CONTINUE
      RETURN
      END


      SUBROUTINE CTYOUT

C     PROVIDE OUTPUT OF RESULTS OF OPT FOR ONE CITY

      COMMON/PAYAR/ PAYZ(100),XZ(100),YZ(100),PAYZT(100) ,IWP
      COMMON/CTYPAR/NAMEC(20),FLATC,FLONGC,FLFCT,TOTPOP,TPX,XSQ,YSQ,
     1SGTX,  SGTY,NTRCTS
      COMMON/RUNVAL/NNEND,LABEL(50), JRAD,IPUNCH,NTYPE ,JPKTP
     1,LSTAPE,NT2,NT3,NT4,NT5,T1,T2,T3,T4,T5
      COMMON/IOAR/ MP,MQ,MS,TA1,TA2,TA3,TA4
C     RECONVERT AIM TO LAT-LONG
  200 DO 201 I=1,IWP
      XZ(I)=XZ(I)/(FLFCT*60.)+FLONGC
      YZ(I)=YZ(I)/60.+FLATC
```

I-110

21

```
  201 CONTINUE
      IF (IPUNCH .EQ. 0) GO TO 10
      WRITE(MS,202) (NAMEC(I),I=1,2)    ,TOTPOP,FLATC,FLONGC,IWP,NTRCTS
 202  FORMAT( 2A4,2X,3F15.6,I5, I8)
      WRITE(MS,203)(J,PAYZT(J), PAYZ(J), XZ(J), YZ(J),   J = 1,IWP)
 203  FORMAT( I5,4F15.5)
 10   CONTINUE
      WRITE(MO,204)(NAMEC(I),I=1,5)    ,TOTPOP,FLATC,FLONGC,IWP,NTRCTS
 204  FORMAT( 1H , 5A4, 3F15.6, I10,I10)
      WRITE(MO,205)(J,PAYZT(J), PAYZ(J), XZ(J), YZ(J),   J = 1,IWP)
 205  FORMAT( 1H ,I5,4F15.6)
      RETURN
      END
```

NEVINS STANDARD

```
      SUBROUTINE OPTWPN(NMSH,NWOPT)

C     NEVINS STANDARD
C     LAST REVISED NOV. 15, 1972
C     GIVEN VALUE TRACTS FOR A CITY AND KILL PROBABILITY TABLE FOR
C     WEAPONS, THIS SUBROUTINE OPTIMIZES THE WEAPON LAYDOWN.
C     THIS SUBROUTINE IS BASED UPON THE PROGRAM DGZSEL DEVELOPED BY
C     H. EVERETT III AND LONG USED BY IDA AND WSEG.
C     NMESH IS THE NUMBER OF MESH POINTS. NWOPT = 1 LISTS EACH WEAPON
C     AS LAID DOWN.


      COMMON/TMPAND/JEAD,NNEND,IPNCHA,IPUNCH,JPKTP,ADJSTF,LSTAPE,LSTC
     1 ,NSP,DESMX,FMAXWP
      COMMON/PKPR/NDS,PK(30),DELPS(30),DSR(30),DELDS(30),BLA(120),DMAX
      COMMON/ST44TA/X(4000),Y(4000),POP(4000),V(4000)
      COMMON/WPNPRB/IWP,XZ(150),YZ(150),PAYZ(150),PAYZT(150)
      COMMON/CITYPR/NAMEC(20),BLD(2),TOTPOP,BLE,FLATC,FLONC,FLFCT,BLF(3)
     1,SGTX,SGTY,BLG(21),NTRCTS
      COMMON/WPNPR/BLW(2)  DEL,BLN(23)
      COMMON/TOPR/BLB,MO,MS,BLC(15)

      MAXWP = FMAXWP
      IWP=1
      PAYTOT=0.
      IF( NWOPT .EQ. 0) GO TO 41
      WRITE(MO, 42) (NAMEC(I), I = 1,20)
 42   FORMAT(1H1,////,1H0,10X, 50HWEAPON BY WEAPON LAYDOWN IN BLAST OPTIM
     1IZATION FOR CITY OF       ,/,30X, 20A4,////)
      WRITE(MO, 43)
 43   FORMAT(1H0, 5H NO.   ,      15H TOTAL VALUE     , 15HVALUE THIS WPN.
     1    , 15H WPN. LANGITUDE    , 15H WPN. LATITUDE     ,/ )
 41   CONTINUE

C     FIND SMALL DISTANCE
      PKTAR = PK(25)/2.
      DO 21 J = 1,NDS
      JJ = NDS - J + 1
      IF(PK(JJ) .GE.PKTAR) GO TO 21
      JUSE = JJ
      GO TO 22
 21   CONTINUE
      JUSE = 1
 22   CONTINUE
      DZER = SQRT(DSR(JUSE))
      STMIN = DZER/H.
```

Y✓

```
C       ADJUST NUMBER OF MESHES TO GIVE MESH SPACING ABOUT EQUAL TO
C       WEAPON RADIUS.
        IF(NMSH .NE. 0) GO TO 23
        TEMP = SQRT(SGTX*SGTY)/DZER
        NMSH = 3.*TEMP
 23     CONTINUE


C       SETUP INITIAL MESH
        TMP = NMSH/3
194     XS = SGTX/TMP
        YS = SGTY/TMP
        XC=0.
        YC=0.
        NX = NMSH
        NY = NMSH
        IF(XS-STMIN) 360,360,362
  360   IF(YS-STMIN) 361,361,362
  361   XS=2.*XS
        YS=2.*YS
        NX = NX/2
        NY = NY /2


  362   CONTINUE
        PAYMAX=0.
C       EVALUATE GRID XC,XS,NX,YC,YS,NY
C       AND SAVE BEST IF BETTER PAYMAX IN XMAX,TMAX
  100   FNY=NY
        FNX=NX
        NNX=2*NX+1
        XAIM=XC-(FNX+1.)*XS
        DO 177 KX=1,NNX
        XAIM=XAIM+XS
        NNY=2*NY+1
        YAIM=YC-(FNY+1.)*YS
        DO 177 KY=1,NNY
        YAIM=YAIM+YS
        JSW=1
        ASSIGN 175 TU NEXT
        GO TO 170
  175   IF(PAY-PAYMAX)  177,177,176
  176   PAYMAX=PAY
        XMAX=XAIM
        YMAX=YAIM
  177   CONTINUE
        IF(XS-STMIN)  178,178,179
  178   IF(YS-STMIN)  190,190,179
  179   XS=XS/3.
        YS=YS/3.
        NX=2
        NY=2
        XC=XMAX
        YC=YMAX
        GO TO 100
  190   PAYZ(IWP)=PAYMAX
        XZ(IWP)=XMAX
        YZ(IWP)=YMAX
        IF(IWP-1)412,412,409
  409   IF(PAYZ(IWP)-PAYZ(IWP-1))412,412,410
  410 IWP=IWP-1
C     REMOVE IWP WEAPON
      XAIM=XZ(IWP)
      YAIM=YZ(IWP)
      JSW=3
      ASSIGN 411 TO NEXT
      GO TO 170
  411 PAYTOT=PAYTOT-PAYR
      GO TO 190
  412 CONTINUE
C     UPDATE
      XAIM=XMAX
      YAIM=YMAX
      ASSIGN 191 TO NEXT
      JSW=2
      GO TO 170
  191 PAYTOT=PAYTOT+PAY
      PAYZ(IWP)=PAY
      PAYZT(IWP) = TOTPOP - PAYTOT

      IF( NWOPT .EQ. 0) GO TO 168
C     PRINT VALUES FOR WEAPON JUST FOUND
      XZP = XZ(IWP)/(FLECT*60.) + FLONGC
      YZP = YZ(IWP) /60. + FLATC
      WRITE(MO,203) IWP, PAYZT(IWP),PAYZ(IWP),XZP,YZP
203   FORMAT(I5, 4F15.6)
168   CONTINUE
      IF(PAY-DESMAX)200,200,192
  192 IF(IWP-MAXWP)  193,200,200
  193 IWP=IWP+1
      GO TO 104
200   CONTINUE


C     WRITE FINAL RESULTS OF WEAPON OPTIMIZATION.
      IF( IPUNCH .EQ. 1 .OR. IPUNCH  .GT. 0) GO TO 45
      RETURN
45    CONTINUE
      IF(IPUNCH .NE. 1) GO TO 46
      WRITE(MS,202)(NAMEC(I),I=1,2)    ,TOTPOP,FLATC,FLONGC,IWP,NTRCTS
202   FORMAT( 2A4,2X,3F15.6,I5, I3)
46    CONTINUE
      IF( IPUNCH.LT. 2) GO TO 47
      WRITE(MO,51) (NAMEC(I),I = 1,20),TOTPOP,FLATC,FLONGC, SGTX,SGTY,
     1 NTRCTS,IWP
51    FORMAT( 1H1 , //////    , 20X, 37HRESULTS OF WEAPON BLAST OPTIMIZAT
     1ION    ,///, 10X,  9H CITY OF    , 20A4, //,  13HCITY POPULATION =,
     2 19H LATITUDE OF CG = , F11.5, 19H LONGITUDE OF CG = ,F11.5,/,
     3 22H E-W POPN STD. DEV. =  ,F10.3,22H N-S POPN STD. DEV. = ,F10.3,
     4  17H NO. OF TRACTS =  ,I6,///, 20X,  I5,18H WEAPONS WERE USED,//)
      WRITE(MO, 43)
47    CONTINUE
      DO 52 J = 1,IWP
      XZP =-XZ(I)/(FLECT*60.) + FLONGC
      YZP = YZ(I)/60. + FLATC
      IF(IPUNCH .NE. 1) GO TO 53
      WRITE(MS,55) J, PAYZT(J),     PAYZ(J), XZP,YZP
55    FORMAT(         I5,4F15.5)
53    CONTINUE
      IF( IPUNCH .LT. 2) GO TO 54
      WRITE(MO,59) J, PAYZT(J),     PAYZ(J), XZP,YZP
```

24

```
56      FORMAT( 1H   , I5,4F15.6)
54      CONTINUF
52      CONTINUF
        RETURN


C       EVALUATF OR UPDATE AS JSW=1, OR 2
C       REMOVE IF JSW IS 3 RECOVERED PAY TO PAYR
C       EXIT TO NEXT
  170   PAY=0.
        PAYR=0.
        DO 161 I=1,NIRCTS
        TEMPX = ABS(XAIM - X(I))
        IF( TEMPX.GT. DMAX) GO TO 161
        TEMPY = ABS(YAIM - Y(I))
        IF(TEMPY .GT.DMAX) GO TO 161
        DD = TEMPX*TEMPX + TEMPY*TEMPY
        J = 1
164     CONTINUF
        J = J + 1
        IF(DSQ(J) .GT. DD) GO TO 164
        P =  PK(J) +(DD - DSQ(J))*DELPS(J)/DELDS(J)
        P = P*DEL
        PAY=PAY+V(I)*P
        GO TO (161,160,400),JSW
  400   VPR=V(I)/(1.-P)
        PAYR=PAYR+VPR-V(I)
        V(I)=VPR
        GO TO 161
  160   V(I)=V(I)*(1.-P)
  161   CONTINUF
        GO TO NEXT,( 175,411,191)
        END
```

Nevuns Standard

```
        SUBROUTINE ONEPAS ( JSW, XAIM,YAIM)

C       NEVUNS STANDARD
C       LAST REVISED NOV. 15, 1972

C       DOES A SINGLE UPDATE TRACT BY TRACT FOR BLAST KILL BY ADDING OR
C       REMOVING A WEAPON

        COMMON/PKPR/BLK,PK(30),DELPS(30),DSQ(30),DELDS(30),BLA(120),DMAX
        COMMON/ST44TA/X(4000),Y(4000),POP(4000),V(4000)
        COMMON/CITYPR/ BLD(55),NTRCTS
        COMMON/WPNPR/BLM(2), DEL,BLN(23)

C       EVALUATF OR UPDATE AS JSW=1, OR 2
C       REMOVE IF JSW IS 3 RECOVERED PAY TO PAYR
  170   PAY=0.
        PAYR=0.
        DO 161 I=1,NTRCTS
        TEMPX = ABS(XAIM - X(I))
        IF( TEMPX.GT. DMAX) GO TO 161
        TEMPY = ABS(YAIM - Y(I))
        IF(TEMPY .GT.DMAX) GO TO 161
        DD = TEMPX*TEMPX + TEMPY*TEMPY
        J = 1
164     CONTINUF
```

25

```
      J = J + 1
      IF(DSQ(J) .GT. DD) GO TO 164
      P =  PK(J) +(DD - DSQ(J))*DELPS(J)/DELDS(J)
      P = P*DEL
      PAY=PAY+V(I)*P
      GO TO (161,160,400),JSW
  400 VPR=V(I)/(1.-P)
      PAYR=PAYR+VPR-V(I)
      V(I)=VPR
      GO TO 161
  160 V(I)=V(I)*(1.-P)
  161 CONTINUE
      RETURN
      END
```

Nevins Standard

```
      SUBROUTINE FLPKED

C     NEVINS STANDARD
C     LAST REVISED NOV. 8. 1972

C     FILLS PK TABLE ACCORDING TO EVERETT FASHION IN ORIGINAL DGZSEL
C     TABLES FILLED FOR BOTH BLAST AND INJURY

      DIMENSION PKC(30),DSQ(30),DELDS(30),DELPS(30)
      COMMON/PKPR/ NDS,PKL(30),DPKL(30),DSQL(30),DDSQL(30),
     1 PKI(30), DPKI(30),DSQI(30),DDSQI(30), DMAX
      COMMON/WPNPR/BLA(3),CEP,BLB(6),NTYPE ,BLC(13),YLDNU,BLD
      COMMON/VULPR/ PSIL,BLK,PSINJ,BLL(3),XMOD,BLE(102)
      COMMON/TRNSFR/ MOD,DZER,SIGD,PTG,PMAX,BLF(15)
      COMMON/IOPR/ BLG,MO,BLH(16)
      EXTERNAL PZZ


      MOD=6

      ZILCH = SSKP(0.,0.,0.,0.,0.,0.)
      ISTRT = 1
      PSI = PSIL
      GO TO 24
23    CONTINUE
      PSI = PSINJ
24    CONTINUE
      IF( NTYPE .LT.0) PSI = - PSI
      DZER = XLPAR(PSI)*YLDNU
      SIGD=CEP/1.1774
      PMAX = SSKP(MOD, DZER,SIGD,SIGD,0.,0.)
      DDMAX=5.*(DZER+CEP)

      DSQ(1)=1.E5
      PKC(1)=0.
      PTG=.001
      DSQ( 2) = RCUTF(0.,DDMAX,.001,PZZ)
      PKC(2)=0.
      PTG=.01
      DSQ( 3) = RCUTF(0.,DDMAX,.001,PZZ)
      PTG=.02
      DSQ( 4) = RCUTF(0.,DDMAX,.001,PZZ)
      PTG=.05
      DO 200 I=5,23
      DSQ( I) = RCUTF(0.,DDMAX,.001,PZZ)
      PTG=PTG+.05
  700 CONTINUE
      PTG=.98
      DSQ(24) = ROUTF(0.,DDMAX,.001,PZZ)
      PTG=.99
      DSQ(25) = ROUTF(0.,DDMAX,.001,PZZ)
      DSQ(26)=0.
      PKC(26)=1.
      DO 701 I=3,25
      XXXX = DSQ(I)
      PKC(I) = SSKP(  MOD,DZER,SIGD,SIGD,XXXX,0.)
      DSQ(I)=DSQ(I)*DSQ(I)
  701 CONTINUE
      DSQ(2)=DSQ(2)*DSQ(2)
      NDS=26
      DMAX=SQRT (DSQ(2))
      DO 710 I = 2,26
      DELDS(I) = DSQ(I) - DSQ(I-1)
      DELPS(I) = PKC(I) - PKC(I - 1)
710   CONTINUE

      IF(ISTRT .NE. 1) GO TO 21
      DO 22 J = 1,NDS
      PKL(J) = PKC(J)
      DPKL(J) = DELPS(J)
      DSQL(J) = DSQ(J)
      DDSQL(J) = DELDS(J)
22    CONTINUE
      ISTRT = 2
      GO TO 23
21    CONTINUE
      DO 26 J = 1,NDS
      PKI(J) = PKC(J)
      DPKI(J) = DELPS(J)
      DSQI(J) = DSQ(J)
      DDSQI(J) = DELDS(J)
26    CONTINUE

      WRITE(MO, 48)
48    FORMAT(////)
      WRITE(MO,8)
8     FORMAT(1H0, 48HVALUES OF PK - DIST BY DGZSEL TYPE CALCULATION     )
      WRITE (MO,43)
43    FORMAT( 1H0,  4H NO. ,9H  PROB L  ,2X, 8H DIST L  ,2X,
     19H  PROB I ,2X, 8H DIST I  , 12X,4H NO.   ,9H  PROB L      ,
     2 2X, 8H DIST L ,2X, 9H  PROB I  ,2X, 8H DIST I  )
      DO 41 I = 1,13
      II = 2*I
      IM = II - 1
      TL1 = SQRT(DSQL(IM))
      TL2 = SQRT(DSQL(II))
      TI1 = SQRT(DSQI(IM))
      TI2 = SQRT(DSQI(II))
      WRITE( MO,42) IM,PKL(IM), TL1,PKI(IM),TI1,
     1 II,PKL(II),IL2, PKI(II),TI2
42    FORMAT( 1H  , 1H(,I2,1H), F9.6, F12.6,F9.6, F12.6,10X,
     11H(,I2, 1H), F9.6,F12.6, F9.6, F12.6      )
41    CONTINUE
      RETURN
      END
```

Nevins Standard

```
      FUNCTION PZZ(XXXX)
C     NEVINS STANDARD
C     LAST REVISED NOV. 7, 1972
C     USED IN CONJUNCTION WITH ROOTF TO FILL A SPECIAL ROLE IN FLPKHU
C     THERE IS NO GENERAL USE FOR THIS FUNCTION.
      COMMON/TRNSFR/MOD,DZER,SIGD.PTG,PMAX,BLA(15)
      EXTERNAL SSKP
      TEMP = SSKP(MOD,DZER,SIGD,SIGD,XXXX,0)
      PZZ = TEMP - PTG*PMAX
      RETURN
      END
```

Nevins Standard

```
      FUNCTION SSKP (MOD.A,SX.SY,XMU,YMU)
C     NEVINS STANDARD
C     LAST REVISED NOV. 7, 1972
C     COMPUTES SINGLE SHOT KILL PROBABILITY FOR ELLIPTICAL NORMAL
C     DISTRIBUTION WITH AIMING ERROR SX AND SY. THE AIMING OFFSET IS
C     XMU AND YMU. A IS THE WEAPON LETHAL RADIUS. MOD IS A SHAPE FACTOR
C     MOD = 1 IS GAUSSIAN IN PK VS ACTUAL DISTANCE FROM THE WEAPON
C     MOD = 3 IS SIG 20, MOD = 6 IS SIG 30 , MOD = INF IS COOKIE CUTTER
C     FOR A DESCRIPTION OF EQUATIONS SEE LAMDA PAPER 6 BY HUGH EVERETT
C      III AND R. GALIANO.
C     THE FUNCTION MUST BE INITIALIZED BY AN INITIAL CALL WITH MOD = 0
C     ZERO VALUE OF SSKP IS RETURNED. AFTERWARDS NORMAL CALLS CAN BE
C     MADE FOR THE DURATION OF THE PROGRAM.
      DIMENSION BIN(250)
      DIMENSION W(11)
      IF(MOD)10,11,10
11    CONTINUE
C     INITIALIZATION
      BIN(1)=1.0
      BIN(2)=1.0
      DO 20 J=2,20
      L=J*(J+1)/2+2
      BIN(L-1)=1.0
      BIN(L-2)=1.0
      NN=J*(J-1)/2+1
      LAST=J-1
      DO 20 K=1,LAST
      BIN(L)=BIN(NN)+BIN(NN+1)
      L=L+1
      NN=NN+1
   20 CONTINUE
      W(1)=2.506628474
      W(2)=2.506628474
      W(3)=7.519885422
      W(4)=37.59942411
      W(5)=263.1959689
      W(6)=2368.763949
      W(7)=26052.40260
      W(8)=338733.23382
      W(9)=5080993.582
      W(10)=86376975.90
      W(11)=1641162542.
      SSKP = 0.
      RETURN
C     NORMAL CALL
   10 C=6.283185*SX*SY
      XMOD=MOD
      B=XMOD/(A*A)
      N=MOD-1
      XJ=1.0
      TSUM=0.
      LL=0
      LLX = LL + 1
      NX = N+1
      DO 7 JX= LLX,NX
      J = JX - 1
      SUM=0.
      NN=J*(J+1)/2+1
      K=0
      KX = K + 1
      JY = J + 1
      DO 3 LX = KX,JY
      L = LX - 1
      NNN=NN+1
      Y1=G(2*L,SX,XMU,B,BIN,W)
      Y2=G(2*(J-L),SY,YMU,B,BIN,W)
      TERM=BIN(NNN)*Y1*Y2
      SUM=SUM+TERM
    3 CONTINUF
      TTERM=B**J*SUM
      IF(J)5,6,5
    5 TTERM=TTERM/XJ
      XJJ=J
      XJ=XJ*(XJJ+1.0)
    6 TSUM=TSUM+TTERM
    7 CONTINUF
      SSKP =TSUM/C
      RETURN
      END

      FUNCTION G(M,SIG,XM,BB,BIN,W)
C     NEVUNS STANDARD
C     LAST REVISED NOV. 7, 1972
C     ONLY USED FOR SPECIAL CALCULATIONS FOR THE FUNCTION SSKP.
C     SEE LAMDA PAPER 6.
      DIMENSION BIN(1),W(11)
      ALPHA=SQRT (1.+2.*BB*SIG*SIG)
      SA=SIG/ALPHA
      IF(XM)11,10,11
   11 BETA=XM/ALPHA
      SB=SIG/BETA
      BA=BETA/ALPHA
      L=0
      SUM=0.0
      NN=M*(M+1)/2+1
```

Nevins Standard

```
      LX = L + 1
      MX = M + 1
      DO 1 KX= LX,MX,2
      K = KX - 1
      NNN=NN+K
      L2=K/2+1
      Z = K
      TERM=BIN(NNN)*S2**Z*W(L2)
      SUM=SUM+TERM
    1 R=EXP ((BETA*BETA-XM*XM)/(2.*SIG*SIG))
      V = M
      G=B1**V*SA**2*SUM
      RETURN
   11 G1=SA**(M+1)
      L3=M/2
      G=G1*W(L3+1)
      RETURN
      END

      FUNCTION XLPAD(C)
C     NEVINS STANDARD
C     LAST REVISED NOV. 7, 1972
C     RETURNS PRESSURE GIVEN DISTANCE. BASED ON OLD WSEG FITS.
      DIMENSION T(13),X(13),Y(26)
      DATA  T( 1) /2000./
      DATA  T( 2) /1000./
      DATA  T( 3) /500./
      DATA  T( 4) /300./
      DATA  T( 5) /100./
      DATA  T( 6) /50./
      DATA  T( 7) /30./
      DATA  T( 8) /20./
      DATA  T( 9) /15./
      DATA  T(10) /10./
      DATA  T(11) /6./
      DATA  T(12) /3./
      DATA  T(13) /1./
      DATA X ( 1) /7.6/
      DATA X ( 2) /6.90776/
      DATA X ( 3) /6.21461/
      DATA X ( 4) /5.70378/
      DATA X ( 5) /4.60517/
      DATA X ( 6) /3.91202/
      DATA X ( 7) /3.40120/
      DATA X ( 8) /2.99573/
      DATA X ( 9) /2.70805/
      DATA X (10) /2.30259/
      DATA X (11) /1.79176/
      DATA X (12) /1.09861/
      DATA X (13) /0./
      DATA Y ( 1) /-2.659/
      DATA Y ( 2) /-1.561/
      DATA Y ( 3) /-1.273/
      DATA Y ( 4) /-1.05/
      DATA Y ( 5) /-.616/
      DATA Y ( 6) /-.301/
      DATA Y ( 7) /-0.062/
```

Nevins Standard
more recent version
available

```
      DATA Y ( 8) /0.1222/
      DATA Y ( 9) /0.44469/
      DATA Y (10) /.77011/
      DATA Y (11) /1.15698/
      DATA Y (12) /1.71919/
      DATA Y (13) /2.2213/
      DATA Y (14) /-2.659/
      DATA Y (15) /-1.561/
      DATA Y (16) /-1.273/
      DATA Y (17) /-1.05/
      DATA Y (18) /-.616/
      DATA Y (19) /-.301/
      DATA Y (20) /-.062/
      DATA Y (21) /.1222/
      DATA Y (22) /.3001/
      DATA Y (23) /.49470/
      DATA Y (24) /.81978/
      DATA Y (25) /1.25276/
      DATA Y (26) /1.88403/
      B2=0
      XX=Q
C     Q GT ZERO IS SURFACE BURST, LESS THAN ZERO OPT. AIR BURST.
      IF (XX.GE.0.)GO TO 1
      B2=13
      XX=-Q
1     TEMP1= XX
      IF (1. .GE.XX) TEMP1=1.001
      DO 2 II=1,13
      I=13-II+1
      IF (T(I) .GT. TEMP1) GO TO 3
2     CONTINUE
      I=12
3     TEMP2= X(I+1)-X(I)
      XXI = I
      B2 = B2 + XXI
      IB = B2
      IBP      = B2 + 1.
      S=(Y(IBP )-Y(IB))/TEMP2
      B=-(S*X(I+1))+Y(IBP )
      Z=ALOG(TEMP1)*S+B
      XLRAD = EXP(Z)
      RETURN
      END

      FUNCTION ROOTF(X0,XF,ERR,DUMMY)
C     NEVUNS STANDARD
C     LAST REVISED NOV. 7, 1972

C     A FUNCTION WHICH RETURNS AS A FUNCTION VALUE THE FIRST ROOT ,
C     USUALLY OF THE FUNCTION DUMMY.
C     USUALLY. OF THE FUNCTION DUMMY.

C     INITITALIZE AMINUS
      ISW=0
      XMINS=X0
      FMINS=DUMMY(AMINS)
      ROOTF=XMINS
      IF(FMINS.EQ.0.)RETURN
      XPLUS=XF
```

Nevins Standard

```
      DELTA = (XF-X0)*2.
      TCON=DELTA*.0005

C     STEP THROUGH UNTIL STRADDLE ROOT.
 1    FPLUS=DUMMY(XPLUS)
      ROOTF=XPLUS
      IF(FPLUS.EQ.0.)RETURN
      IF(FPLUS*FMINS.LT.0.)GO TO 2
      FMINS=FPLUS
      XMINS=XPLUS
      XPLUS = XPLUS + DELTA
      IF(XPLUS-XF.LT.0.)GO TO 1
C     DECREASE DELTA AND START OVER LOOKING FOR ROOT.
      DELTA=DELTA*.5
      XPLUS=DELTA*.5+X0
      IF(DELTA-TCON.GE.0.) GO TO 1
C     ERROR EXIT
 3    ROOTF=XF+1.
      RETURN
 2    IF(FPLUS .GE. 0.) GO TO 11
      TEMP=XPLUS
      XPLUS=XMINS
      XMINS=TEMP
      TEMP=FPLUS
      FPLUS=FMINS
      FMINS=TEMP
 11   IF(ISW .EQ. 0) GO TO 12
      ISW=0
C     ILINP-LINEAR INTERPOLATION
      T1=FPLUS-FMINS
      XP=(XMINS-XPLUS)*FMINS/T1+XMINS
      GO TO 13
C     IBNRY-BINARY DIVIDE
 12   XP=(XPLUS+XMINS)*.5
      ISW=1
 13   FP=DUMMY(XP)
      ROOTF=XP
      IF(FP .EQ. 0.)RETURN
      IF( FP .LT. 0.) GO TO 14
      FPLUS=FP
      XPLUS = XP
      GO TO 15
 14   FMINS=FP
      XMINS=XP
 15   IF(XMINS+XPLUS .EQ. 0.) GO TO 11
      T1=XMINS+XPLUS
      TEMP = (XMINS-XPLUS)/T1
      IF(TEMP .GE. 0.) GO TO 16
      TEMP = - TEMP
 16   IF (TEMP-ERR .GE.0.)GO TO 11
      ROOTF=(XMINS+XPLUS)*.5
      RETURN
      END
```

Nevins Standard

```
      SUBROUTINE CALRN

C     NEVINS STANDARD
C     LAST REVISED NOV. 2[illegible], 1972
```

```
C... THE PURPOSE OF THIS SUBROUTINE IS TO PUT RANDOM NUMBER
C      GENERATOR CALLS IN ONE PLACE
C      THIS WILL ALLOW CONVERTING TO OTHER SYSTEMS WITH A MINIMUM OF PAIN
C... IF RNO = ZERO GET A NEW RANDOM NUMBER. -INSERT INTO RNW
C      IF RNO IS POSITIVE OBTAIN SEED FOR RANDOM NUMBERS. THIS MUST
C      BE DONE BEFORE THE FIRST RANDOM NUMBER IS GENERATED.
C      IF RNO = 1. OBTAIN SEED (RANDOMLY) BY READING CLOCK,OTHERWISE
C      USE RNO AS VALUE OF SEED.
C      IF RNO IS -1 DRAW NUMBER FROM EXPONENTIAL DISTRIBUTION, ANY OTHER
C      NEGATIVE NUMBER DRAW FROM NORMAL DISTRIBUTION, N(0,1)

       COMMON /TRAN/RNO,RNW

       IF(RNO .NE. 0.) GO TO 10

C      GET A RANDOM NUMBER UNIFORMLY DISTRIBUTED FROM 0 TO 1,FLOATING PT.
C      RANF(0) IS UNIQUE TO CDC 6400 FTN
       RNW = RANF(0)
       RETURN

10     CONTINUE
       IF (RNO .LT. 0.)GO TO 20
       IF(RNO .EQ.1.)GO TO 15

C      INITIALIZE RANDOM NUMBER GENERATOR WITH RNO AS SEED
C      RANSET IS UNIQUE TO CDC 6400 FTN
       CALL RANSET(RNO)
       RETURN

C      READ THE SOFTWARE CLOCK TO OBTAIN SEED HOPEFULLY AT RANDOM.
C      THE FUNCTION TIME IS CDC 6400 FTN UNIQUE.
15     CONTINUE
       CALL TIME(CLTIM)
       CLTIM = ABS(CLTIM)
       RNI = CLTIM + 1.2
       CALL RANSET(RNI)
       RETURN

20     CONTINUE
       IF(RNO .NE. -1.) GO TO 40

C      DRAW NUMBER FROM EXPONENTIAL DISTRIBUTION WITH MEAN = 1
       U = RANF(0)
       RNW = - ALOG(U)
       RETURN

40     CONTINUE

C      DRAW NUMBER FROM NORMAL DISTRIBUTION WITH MEAN 0 AND STD. DEV. 1
C      USE AS METHOD ALGORITHM TR FROM *COMPUTER METHODS FOR SAMPLING
C      FROM THE EXPONENTIAL AND NORMAL DISTRIBUTIONS*, J. H. AHERNS
C      AND U. DIETER, COMMUNICATIONS OF THE ACM, OCT., 1972, VOL. 15,
C      NO. 10, P. 873.

41     CONTINUE
       U = RANF(0)
       U0 = RANF(0)
       IF(U .GE. 0.919544405706926 ) GO TO 42
       RNW = 2.40375765693742*(U0 + U*0.825339282536923) -
```

```
     1 2.1140280333742

42      CONTINUE
        IF( U .LT. 0.9654871312138581 GO TO 44

43      CONTINUE
        U1 = RANF(0)
        Y = 4.4691147371392 7 - 2.*ALOG(U1)
        U2 = RANF(0)
        IF( Y*U2 .GT. 2.1140280833372 4) GO TO 43
        GO TO 49

44      CONTINUE
        IF(U.LT. 0.9449907084733023) GO TO 46

45      CONTINUE
        U1 = RANF(0)
        Y = 1.0463997474377 71 + U1*0.2730273359397 06
        U2 = RANF(0)
        TEMP = 0.3989422804014 33*EXP(-Y*Y/2.) - 0.443299125820220
       1 + Y*0.2096440571954 86
        IF( TEMP .LT. U2*0.0427025314907 35) GO TO 45
        GO TO 49

46      CONTINUE
        IF( U .LT. 0.9258523337577 14 ) GO TO 48

47      CONTINUE
        U1 = RANF(0)
        Y = 0.2097297736 30000 + U1*1.5506691737977 1
        U2 = RANF(0)
        TEMP =  0.3989422804014 33*EXP(-Y*Y/2.) - 0.443299125820220
       1 + Y* 0.2096440571954 86
        IF( TEMP .LT. U2*0.0159745224652 38) GO TO 47
        GO TO 49

48      CONTINUE
        U1 = RANF(0)
        Y = U1* 0.2097297360000 0
        U2 = RANF(0)
        TEMP = 0.3989422804014 33*EXP(-Y*Y/2.) - 0.3825445560425 18
        IF(TEMP .LT. U2* 0.0103977243584 19) GO TO 48
        GO TO 49

49      CONTINUE
        IF( U1 .LT. 0.5) GO TO 50

        RNW = -Y
        RETURN
50      CONTINUE
        RNW = Y
        RETURN
        END
```

NEVINS Standard

```
        FUNCTION CUMNOR (X)

C       NEVINS STANDARD
C       LAST REVISED NOV. 7, 1972

C       COMPUTES Y = P(X) = PROBABILITY THAT THE RANDOM VARIABLE U,
C     DISTRIBUTED NORMALLY(0,1), IS LESS THAN OR EQUAL TO X.
C     IF X IS GREATER THAN 7  P = 1
C     IF X IS LESS THAN -7  P = 0
C     SEE HASTINGS APPROX FOR DIGITAL COMPUTEERS. P. 169
C     BETTER IS HASTINGS P. 187 WHICH DOES NOT HAVE TO USE THE EXP
C     FUNCTION. SEE FUNCTION CUMNOA

      AX = ABS(X)
      IF (AX - 7.0) 10,10,20
   10 CONTINUE
      T = 1.0/(1.0+.2316419*AX)
      D = 0.3989423*EXP(-0.5*X*X)
      P = 1.0 - D*T*((((1.330274*T - 1.821256)*T + 1.781478)*T -
     A    0.3565638)*T + 0.3193815)
      IF (X) 1,2,2
    1 P = 1.0 - P
    2 CUMNOR = P
      RETURN
   20 IF (X) 30,30,40
30    CUMNOR=0.
      RETURN
40    CUMNOR=1.0
      RETURN
      END
```

Nevuns Standard

```
      SUBROUTINE PDIST(NTYPE,LOGV,PRES,DIST)

C     NEVUNS STANDARD
C     LAST REVISION OCT. 30, 1972

C     COMPUTES DISTANCE GIVEN PRESSURE
C     DISTANCE  IS IS NAUTICZL MILES. PRESSURE IS PSI. ONE MT. WEAPON
C     YIELD IS ASSUMED. BASED ON A FIT TO HEIGHT OF BURST CURVES IN
C     EFFECTS OF NEUCLEAR WEAPONS. THE SAME EQUATIONS ARE USED FOR THE
C     INVERSE CALCULATION IN THE SUBROUTINE PROMPT.
C     FIT TO WITHIN 1% PER CENT FROM 1 TO 200 PSI

C     NTYPE = 1 IS SURFACE, EQUAL 0 IS 10PSI OPTIMIZED,=-1 IS OPT
C     AIRBURST
C     IF LOGV = 0 CALL WITH PRESSURE, IF 1 CALL WITH LOG PRESS


      IF (LOGV .EQ. 1) GO TO 6
      IF (PRES .GT. 0.00001) GO TO 9
      DIST = 99999.
      RETURN
6     CONTINUE
      IF( PRES .GT. -5.) GO TO 9
      DIST = 99999.
      RETURN
9     CONTINUE

      IF( NTYPE .LE.0) GO TO 20

C     SURFACE BURST
      IF( LOGV .EQ. 1) GO TO 11
      XLPRES = ALOG10(PRES)
      GO TO 12
11    CONTINUE
      XLPRES = PRES
```

```
12    CONTINUE
      IF( XLPRES .LE. 1.3010) GO TO 13
      SLOPE = -2.255
      GO TO 14
13    CONTINUE
      SLOPE = -1.865
14    CONTINUE
      XLDIST = 0.0546 + ( XLPRES - 1.3010)/SLOPE
      DIST = 10.**XLDIST
      RETURN
20    CONTINUE
      IF( NTYPE .LT. 0) GO TO 50

C     10 PSI OPT AIRBURST
      PRE = PRES
      IF( LOGV .EQ.1) GO TO 21
      IF( PRES .GT. 15.)     GO TO 30
      XLPRES = ALOG10(PRES)
      GO TO 40
21    CONTINUE
      XLPRES = PRES
      IF( XLPRES .LT. 1.1761) GO TO 40
      PRE  = 10.**XLPRES
      GO TO 30
30    CONTINUE
C     NO ERROR EXIT IF PRESSURE GREATER THAN 38PSI
      IF( PRE  .LE. 38.) GO TO 31
      DIST = 0.
      RETURN
31    CONTINUE
      IF( PRE .LT.28.) GO TO 32
      DIST = ( 38. - PRE )/19.5
      RETURN
32    CONTINUE
      DIST = 3.922 + ( 28. - PRE )/5.1
      RETURN

40    CONTINUE
C     LOW PRESSURE
      XLDIST = .2737 + (1.1761- XLPRES)/1.75
      DIST = 10.**XLDIST
      RETURN

C     OPT AIRBURST
50    CONTINUE
      IF( LOGV .EQ. 1)  GO TO 51
      XLPRES = ALOG10(PRES)
      GO TO 55
51    CONTINUE
      XLPRES = PRES
55    CONTINUE
      IF( XLPRES .GT. 1.30103) GO TO 52
      SLOPE = -1.52
      GO TO 53
52    CONTINUE
      SLOPE = -1.330
53    CONTINUE
      XLDIST = .1903 + ( XLPRES - 1.3010)/SLOPE
      DIST = 10.**XLDIST
      RETURN
      END
```

Nevins Standard
More recent version available

```
      SUBROUTINE PROMPT
C     NEVINS STANDARD
C     LAST REVISED OCT. 31, 1972
C      TO CALCULATE PRESSURE(PSI), THERMAL RADIATION(CAL/CM2),
C      AND INITIAL NUCLEAR *RADIATION(RAD) GIVEN DISTANCE(N MI.)
C    NTYPE = 1 SURFACE BURST*  0 IS 10PSI OPTIMIZED AIRBURST*
C     IF -1 IS A OPTIMUN AIRBURST HEIGHT
C     LONG = 1 DO WHOLE CALCULATION, 0 DO PRESSURE ONLY
C     LONG = 2 DO SPECIALIZED INR CALCULATION.

      COMMON/EFFCAL/ YIELD,YLDPCR, YLDPLG, LONG, NTYPE,DSTP, HOB,PRES,
     1 XLPRES,THER, THRPVS,  RAD,AR, BLANK(12)
      DATA PMIN, PMAX, XLPMIN,XLPMAX/0.001, 99999., -3., 5./
      DATA THRYF/ 0.333333/


      DIST = DSTP/YLDPCR
      IF( DIST .GT. 0.00001) GO TO  4
      DIST = 0.00001
4     CONTINUE
      IF ( NTYPE .LE. 0) GO TO 40

C     SURFACE BURST
      XLDIST = ALOG10(DIST)
      IF( DIST .GE. 1.147) GO TO 2
1      SLOPE = -2.255
       GO TO 3
2     CONTINUE
      SLOPE = - 1.825
3      CONTINUE
       XLPRES = 1.3010  + SLOPE*(XLDIST - 0.0596)
       PRES = 10.** XLPRES
      GO TO 60
40    CONTINUE
      IF( NTYPE .LT. 0)  GO TO 50

C     10 PSI OPTIMIZED BURST
      IF( DIST .LT. 1.90) GO TO 41
      XLDIST = ALOG10(DIST)
      XLPRES = 1.1761 - 1.75*(XLDIST - .2787)
      PRES = 10.**XLPRES
      GO TO 60
41    CONTINUE
      IF( DIST.LT. 0.922) GO TO 42
      PRES = 20. - 5.1*(DIST - .922)
      XLPRES = ALOG10(PRES)
      GO TO 60
42    CONTINUE
      PRES = 38. - 19.5*DIST
      XLPRES = ALOG10( PRES)
      GO TO 60
50    CONTINUE

C     OPTIMIZED AIRBURST HEIGHT
      XLDIST = ALOG10(DIST)
      IF(DIST .LT. 1.55) GO TO 52
      SLOPE = -1.52
      GO TO 53
52    CONTINUE
      SLOPE = - 1.830
53    CONTINUE
      XLPRES =  1.3010 + SLOPE*(XLDIST -  .1903)
      PRES = 10.**XLPRES
60    CONTINUE
       IF( PRES .LT. PMIN) PRES = PMIN
       IF( PRES .LT. PMIN) XLPRES = XLPMIN
       IF(PRES .GT. PMAX) PRES = PMAX
       IF( PRES .GT. PMAX) XLPRES = XLPMAX
      IF( LONG .NE. 0) GO TO 72
      RETURN


C     NOW DO NUEC RAD
72    CONTINUE
      IF( LONG .EQ.2) GO TO 80

C     NORMAL INR CALCULATION
      IF( NTYPE .EQ. 1) GO TO 70
C     10 PSI OR OPT AIR BURST
      CRD = 7.25E+12
      CRDH = 3.62E-3
      SRS = 5.07E+6 + 3.42E+6*DSTP*DSTP
      SR = SQRT(SRS)
      GO TO 71
70    CONTINUE
C     SURFACE BURST
      CRD = 3.2E+12
      CRDH = 3.35E-3
      SR = 1850.*DSTP
      SRS = SR*SR
71    CONTINUE
      ARD = YIELD*(CRD - 3.2E+12*YLOPLG)
      RRD = CRDH - 0.92E-3*YLOPLG
      RAD = ARD*EXP(-RRD*SR) /SRS
      IF( RAD .GT. 1.E+8) RAD = 1.E+8
      GO TO 11

80    CONTINUE
C     SPECIALIZED INR CALCULATION
C     TO OBTAIN INR DIRECTLY FROM SLANT RANGE
C     AR IS VALUE OF RHO IN THIS CALCULATION
      IF (NTYPE.EQ.0) GO TO 81
C     SURFACE BURST
C     THIS CANNOT HANDLE AIR BURSTS OTHER THAN 10 PSI OPT
      SLNTR = DIST*DIST + .0029119
      SLNTR = SQRT(SLNTR)
      GO TO 85
81    CONTINUE
C     10 PSI OPT
      IF( YIELD .EQ. 1.) GO TO 83
      IF (YIELD .EQ. .3)GO TO 84
C     40 KT
      SLNTR = DIST*DIST + 0.178916
      SLNTR = SQRT(SLNTR)
      GO TO 85
83    CONTINUE
      SLNTR = DIST*DIST + 1.4612
      SLNTR = SQRT(SLNTR)
      GO TO 82
84    CONTINUE
      SLNTR = DIST*DIST + 0.65477
      SLNTR = SQRT(SLNTR)
82    CONTINUE
      IF( YIELD .EQ. .04) GO TO 91
      IF( YIELD .EQ. .3) GO TO 92
C     YIELD IS 1MT NOW
      IF( AR .EQ. 1.1) GO TO 93
      IF( AR.EQ. 1.2) GO TO 94
      SLOPE = -4.0733
      SRATE = 0.8920
      GO TO 95
93    SLOPE = -3.4638
      SRATE =  1.0107
      GO TO 95
94    SLOPE = -3.7821
      SRATE = .9487
      GO TO 95
C     YIELD = .3
92    SLOPE = -4.5362
      SRATE =  0.7274
95    CONTINUE
      XLGRAD = 4. + SLOPE*(SLNTR - SRATE)
      GO TO 120
91    CONTINUE
C     YIELD = 40 KT
      IF( AR.EQ. 1.1) GO TO 101
      IF(AR .EQ. 1.2) GO TO 102
C     RHO = 1.3
      IF(SLNTR .GT.0.29139) GO TO 103
      SLOPE = -21.1274
      SRVAL = 0.29139
      VALLG = 5.14613
      GO TO 110
103   CONTINUE
      SLOPE = -6.0259
      SRVAL = .4803
      VALLG = 4.
      GO TO 110
102   CONTINUE
      IF( SLNTR .GT. 0.29949) GO TO 105
      SLOPE = -17.9641
      SRVAL = 0.29949
      VALLG = 5.14613
      GO TO 110
105   CONTINUE
      SLOPE = -5.5741
      SRVAL = .5072
      VALLG = 4.
      GO TO 110
101   CONTINUE
      IF(SLNTR .GT. 0.31298) GO TO 106
      SLOPE = -14.0806
      SRVAL = 0.31298
      VALLG = 5.14613
      GO TO 110
106   CONTINUE
      SLOPE = -4.7824
      SRVAL = 0.5369
      YALLG = 4.
110   CONTINUE
      XLGRAD = YALLG + SLOPE*(SLNTH - SRVAL)
120   CONTINUE
      IF( XLGRAD .GT. 8.) GO TO 121
      RAD =  10.**XLGRAD
      GO TO 11
121   RAD = 1.E+8
11     CONTINUE

C      NOW THERMAL RADIATION
      THER = 7.94E+4*THRYF*YIELD *EXP(- THRPVS*SR)/SRS
      IF( THER .GT. 1.E+8) THER = 1.E+8
      RETURN
       END
```

Nevens Standard

```
      SUBROUTINE FLPK

C     NEVINS STANDARD
C     LAST REVISED ON NOV. 9, 1972

C     FILLS THE ARRAYS DSOI,PKCI,DSOK,PKCK,DELDSI,DELPSI,DELDSK,DELPSK
C     WITH PRESSURE PK DISTANCE RELATIONSHIPS DIRECTLY CALCULATED
C     THE EFFECTS OF CEP ARE INCLUDED BY DIRECT INTEGRATION OF PROBS

      DIMENSION PKT(26) , RLGPP(26),DRL(26),DRC(26)
      COMMON/TMPAND/JREAD,NNEND,IPNCHA,IPUNCH,JPKTP,ADJSTF,LSTAPE,LSTC
     1 ,NSP,DESMX,RMAXWP
      COMMON/DKPR/NDS,PKCK(30),DELDSK(30), DSOK(30),DELDSK(30),
     1 PKCI(30),DELPSI(30), DSOI(30), DELDSI(30),DMAX
      COMMON/VULPR/PSI,SIGBL,PSINJ,SIGBC,BLANK(103)
      COMMON/WPNPR/BLA(3),CEP,RLB(6),NTYPE ,RLC(13),YLDND,BLD
      COMMON/EFFCAL/RLE,YLDDD,RLF,JTIMP,JHTPR,DSTP,RLG,PRESS,PRELP,
     1 RLH(10)
      COMMON/TOPR/MP,MJ,MS,BNA(16)

      DATA PKT/.0001,.001,.01,.02,.05,.10,.15,.20,.25,.30,.35,.40,.45,
     1 .50,.55,.60,.65,.70,.75,.80,.85,.90,.95,.98,.99,.9999/
      DATA RLGPP/-4.4170,-3.2905,-2.32635,-2.05375,-1.64475,-1.28155,
     1 -1.03643,-0.84162,-0.67449,-0.52440,-0.38532,-0.25335,-0.12566,
     2 0.0,0.12566,0.25335,0.38532,0.52440,0.67449,0.84162,1.03643,
     3 1.28155,1.64475,2.05375,2.32635,4.4170/

      IF(JPKTP .NE. 2) GO TO 1
C     READ DATA INTO ARRAYS AND DO NOT CALCULATE IT.
      DO 5 K = 1,26
      READ(MP  ,52)J,DSOI(J),PKCI(J)   ,DSOK(J),PKCK(J) ,J,DELDSI(J),
     1DELPSI(J),DELDSK(J) ,DELPSK(J)
5     CONTINUE
      RETURN
1     CONTINUE

C     TO USE IN /EFFCAL/
      YLDDD = YLDND
      JHTPR = NTYPE
      JTIMP = 0
```

40

```
C     FIRST FIND VALUES OF PRESSURE AND THEN DISTANCE FOR THAT PRESS.
      RLPM = ALOG10(PSI)
      RCPM = ALOG10(PSINJ)
      SIGPL = RLPM*SIGBL
      SIGPC = RCPM*SIGBC
      DO 10 J = 1,26
      RLL = RLGPP(J)*SIGBL*RLPM + RLPM
      IF( NTYPE .NE. 0) GO TO 7
      IF( RLL .GT. 1.57978) GO TO 6
      GO TO 7
6     DIST = 0.
      GO TO 8
7     CONTINUE
      CALL PDIST(NTYPE,1,RLL,DIST)
      DIST = DIST*YLDNU
8     CONTINUE
      DBL(J) = DIST
      DSQK(J) = DIST*DIST
      RCL = RLGPP(J) *SIGBC* RCPM + RCPM
      IF (NTYPE .NE. 0) GO TO 3
      IF( RCL.GT. 1.57978)  GO TO 2
      GO TO 3
2     DIST = 0.
      GO TO 4
3     CONTINUE
      CALL PDIST(NTYPE,1,RCL,DIST)
      DIST = DIST*YLDNU
4     CONTINUE
      DBC(J) = DIST
      DSQI(J) = DIST*DIST
10    CONTINUE

C     OUTPUT INITIAL CALCULATIONS.
      WRITE(MO,15)
15    FORMAT(1H1)
      WRITE(MO,16)
16    FORMAT(///////)
      WRITE (MO,11)
11    FORMAT( 1H0, 36HZERO CEP PK DIST LETHAL AND INJURY     )
      WRITE (MO,12)
12    FORMAT( 1H0, 4H NO., 8H  PROB    ,2X, 8H DIST L  ,4X,8H DIST I
     1,12X  , 4H NO.  , 8H  PROB   ,2X,   8H DIST L  ,4X, 8H DIST I    )
      DO 14 I = 1,13
      II = 2*I
      IM = II - 1
      WRITE(MO,13) IM,PKT(IM) , DBL(IM)  ,DBC(IM) ,II,PKT(II),
     1DBL(II) ,DBC(II)
13    FORMAT(1H  ,1H(, I2,1H), F8.6,2F12.6,10X,1H(,I2,1H),
     1F8.6,    2F12.6)
14    CONTINUE

C     ADJUST DISTANCES TO GET A BETTER INTERPOLATION TABLE.
      DSQK(1) = 0.79012344E+10
      DBL(1) = 8888.8888
      DSQI(1) = 0.79012344E+10
      DBC(1) = 8888.8888
      DBL(24) = 0.666667*DBL(23)
      DBL(25) = 0.333333*DBL(23)
      DBC(24) = 0.666667*DBC(23)
      DBC(25) = 0.333333*DBC(23)
      DSQK(24) = DBL(24)*DBL(24)
```

41

```
      DSQR(25) = DBL(25)*DBL(25)
      DSQI(24) = DBC(24)*DBC(24)
      DSQI(25) = DBC(25)*DBC(25)
      DBL(26) = 0.
      DBC(26) = 0.
      DSQR(26) = 0.
      DSQI(26) = 0.
      SIG = CEP/1.1774
      DBL(2) = DBL(2) + 4.*SIG
      DBC(2) = DBC(2) + 4.*SIG
      DSQR(2) = DBL(2)*DBL(2)
      DSQI(2) = DBC(2)*DBC(2)

C     SETUP FOR NUMERICAL INTEGRATION FOR CEP EFFECTS.
      DIL = SIG/2.5
      SIGS = SIG*SIG
      TPSIGS =1./(2.*3.14159265*SIGS)
      TSIGS = 1./(2.*SIGS)

C     INTEGRATE FOR EACH DISTANCE
      DO 100 JK = 3,24
      SUMCL = 0.
      SUMCC = 0.
      SUMCP = 0.
      CROSS = -DIL
C     CTRC AND CTRR USED AS WEIGHTS IN SIMPSONS RULE
      CTRC = 1.

      DO 20 K = 1,21
      CROSS = CROSS + DIL
      CTRR = 1.
      DOWN = -11.*DIL
      SUMRL = 0.
      SUMRC = 0.
      SUMRP = 0.

      DO 30 J = 1,21
      DOWN = DOWN + DIL
      RAD = CROSS*CROSS + DOWN*DOWN
      PROBD = TPSIGS*EXP(-RAD*TSIGS)
      T1 = DBL(JK) - DOWN
      T2 = T1*T1 + CROSS*CROSS
      DSTP = SQRT(T2)
      CALL PROMPT
      XLPL = PRELP
      T1 = DBC(JK) - DOWN
      T2 = T1*T1 + CROSS*CROSS
      DSTP = SQRT(T2)
      CALL PROMPT
      XLPC = PRELP
      T1 = ( XLPL - ALPM)/SIGPL
      PROL = CUMNOR(T1)
      T2 =  (XLPC - ACPM)/SIGPC
      PROC = CUMNOR(T2)
      IF( J.EQ. 21) CTRR = 1.
      SUMRL = SUMRL + CTRR*PROBD*PROL
      SUMRC = SUMRC + CTRR*PROBD*PROC
      SUMRP = SUMRP + CTRR*PROBD
      IF( CTRR - 2.) 31,31,32
31    CONTINUE
      CTRR = 4.
```

4✓

```
      GO TO 33
32    CONTINUF
      CTRP = 2.
33    CONTINUF
30    CONTINUF
      IF( K.EQ. 21 ) CTRC = 1.
      SUMCL = SUMCL + CTRC*SUMRL
      SUMCC = SUMCC + CTRC*SUMRC
      SUMCP = SUMCP + CTRC*SUMRP
      IF( CTRC - 2.) 21,21,22
21    CONTINUF
      CTRC = 4.
      GO TO 23
22    CONTINUF
      CTRC = 2.
23    CONTINUF
20    CONTINUF
C     PDELN IS USED TO NORMALIZE PROBABILITY INTEGRAL SINCE INTEGRATION
C     IS NOT EXACT
      PDELN = 2.*SUMCP*DIL*DIL/9.
      PKCK(JK) = 2.*SUMCL *DIL*DIL/(9.*PDELN)
      PKCI( JK) = 2.*SUMCC*DIL*DIL/(9.*PDELN)
100   CONTINUF

C     ADJUSTMENTS FOR INTERPOLATION  TABLE SINCE DBL(1) IS LARGE.
      PKCI(1) = 0.
      PKCK(1) = 0.
      PKCI(2) = 0.
      PKCK(2) = 0.

C.... INTEGRATE OVER LETHAL AREA TO NORMALIZE KILL FUNCTIONS
      TPS1 = 3.14159265
      TPS2 = 0.5*3.14159265
      SINTK = 0.
      SINTI = 0.
      DO 70 I = 3,26
      BKI = (PKCK(I) - PKCK(I - 1))/ (DSQK(I) - DSQK(I - 1))
      AKI = PKCK(I -1) - DSQK( I-1)*BKI
      BII = (PKCI(I)- PKCI(I-1) )/(DSQI(I  ) -DSQI(I-1)    )
      AII = PKCI(I-1)-DSQI(I-1)*BII
      SINTK = SINTK -TPS1*AKI*( DSQK(I)        - DSQK(I-1)             )
     1 - TPS2*BKI*( DSQK(I)*DSQK(I)        - DSQK(I-1)*DSQK(I-1) )
      SINTI = SINTI -TPS1*AII*(          DSQI(I) - DSQI(I-1)   )
     1-TPS2*BII*( DSQI(I)*DSQI(I)         -DSQI(I-1)*DSQI(I-1)   )
70    CONTINUF
      AKN = 3.14159265*DSQK(14)
      AIN = 3.14159265*DSQI(14)
      RAK = AKN/SINTK
      SRAK = SQRT(RAK)
      RAI = AIN/SINTI
      SRAI = SQRT(RAI)

      WRITE (MQ,71) SRAK,SRAI,ADJSTF
71    FORMAT(1H0,53HRATIO OF LETHAL RADIUS AREA TO CEP INTEGRATED AREA I
     1S   ,F10.5 ,21H FOR FATALITIES, AND ,F10.5,13H FOR INJURIES,
     2 /,27H DISTANCES ARE ADJUSTED BY  ,F5.3, 14H OF THIS RATIO
     3, 13H TO NORMALIZE  )
      DO 72 J = 1,26
      DSQK(J) = DSQK(J)*( 1. +(RAK-1.)*ADJSTF)
      DBL(J) = DBL(J)*( 1. + (SRAK-1.)*ADJSTF)
      DSQI(J) = DSQI(J)*( 1. + (RAI -1.)*ADJSTF)
      DBC(J) = DBC(J)*( 1.+ (SRAI - 1.)*ADJSTF)
72    CONTINUE
      WRITE(MO,17)
17    FORMAT(//)
      WRITE( MQ,44)
44    FORMAT( 1H0, 42HCEP INTEGRATED PK DIST LETHAL AND INJURY         )
      WRITE (MQ,43)
43    FORMAT( 1H0,  4H NO. ,9H  PROB L   ,2X, 8H DIST L   ,2X,
     19H  PROB I ,2X, 8H DIST I   , 12X,4H NO.   ,9H  PROB L       ,
     2 2X, 8H DIST L ,2X, 9H  PROB I   ,2X, 8H DIST I  )
      DO 41 I = 1,13
      II = 2*I
      IM = II - 1
      WRITE (MQ, 42) IM,PKCK(IM), DBL(IM), PKCI(IM), DBC(IM),
     1II,PKCK(II),DBL(II), PKCI(II), DBC(II)
42    FORMAT( 1H  , 1H(,I2,1H), F9.6, F12.6,F9.6, F12.6,10X,
     11H(,I2, 1H), F9.6,F12.6, F9.6, F12.6       )
41    CONTINUE
C     FILL DIFFERENCE TABLE
      DO 61 I = 2,20
      DELDSI(I) = DSQI(I) - DSQI(I - 1)
      DELPSI(I) =  PKCI(I) - PKCI(I - 1)
      DELDSK(I) = DSQK(I ) - DSQK(I - 1)
      DELPSK(I) = PKCK(I) - PKCK(I - 1)
61    CONTINUE
      DMAX = DBL(2)

      IF (IPNCHA .NE. 1) GO TO 51
C     PUNCHED CARD OUTPUT
      DO 53 J = 1,20
      WRITE( MS,52)J,DSQI(J),PKCI(J)   ,DSQK(J),PKCK(J)  ,J,DELDSI(J),
     1DELPSI(J),DELDSK(J) ,DELPSK(J)
52    FORMAT(I4,4E15.9,/, I4,4E15.9 )
53    CONTINUE
51    CONTINUE
      RETURN
      END
```

COMPUTER PROGRAM DESCRIPTION

NAME: GEM/PADECON

SYNOPSIS: General Economic Model to Assess Recovery from Nuclear Attack

TYPE: PRODUCTION

USE: Equilibrium Economic Model--Part of MEVUNS system

BACKGROUND: This program is the result of an extensive economic modeling effort

DESCRIPTION: A large economic model which includes demand prediction, supply calculations, production functions, capital accretion, inventory, bottleneck calculations, etc.

INPUT: IDA Tapes 795 + 691
Cards

OUTPUT: IDA Runs #69
Separate group of runs

STORAGE: IDA Card Deck 102
IDA Tape #767

DOCUMENTATION: IDA Report S-394 "Methodology for Evaluating the Vulnerabilities of National Systems, Vol. I, Part I, Description of Methodologies," J. McGill, et al., November 1971

LANGUAGE/SYSTEM:
FTN/6400 SCOPE

COMMENTS: A second version of the basic convergence subroutine (COBWEB) is IDA Card Deck #103.

Preceding Page Blank - NOT FILMed

COMPUTER PROGRAM DESCRIPTION

NAME: TOGEM

SYNOPSIS: MEVUNS Auxiliary Program

TYPE: Production

USE: Convert ANCET mortalities to GEM input format

BACKGROUND: Developed as part of GEM

DESCRIPTION: ANCET prompt effects mortalities are converted to GEM input on a countrywide allocation of damage

INPUT: ANCET Output Tape
Geographic Economic Data

OUTPUT: Punched Cards for GEM input

STORAGE: IDA Card Deck #165

DOCUMENTATION: IDA Study S-394, "Methodologies for Evaluating the Vulnerabilities of National Systems," Vol. I, Part I, Description of Methodologies," J. McGill, et al., Nov 1971

LANGUAGE/SYSTEM:
FTN/6400 SCOPE

COMMENTS:

COMPUTER PROGRAM DESCRIPTION

NAME: GEMLET

SYNOPSIS: Stripped version of GEM

TYPE: Test

USE: To run parametric variations of the GEM program to attempt to assess the latter's accuracy. Limited to 10 sectors.

BACKGROUND: Developed in 1974 from the basic GEM program. Much faster running and simpler to use version

DESCRIPTION: Contains basic GEM algorithms but a stripped down set of input requirements.

INPUT: Parameter cards

OUTPUT: IDA Runs #55 -- 20 runs
#57 -- 24 runs

STORAGE: IDA Card Deck #100
IDA Tape #733

DOCUMENTATION: "The Economic Recovery Model," R. Michaels, December 11, 1973

LANGUAGE/SYSTEM: FTN/6400 SCOPE

COMMENTS:

COMPUTER PROGRAM DESCRIPTION

NAME: MARATHON

SYNOPSIS: ABM Deployment Analysis

TYPE: Production

USE: Compare optimized ABM and shelter deployments

BACKGROUND: Developed by E. Pearsall to provide a method of analyzing the benefits of an optimized shelter on ABM deployment against an optimized attack.

DESCRIPTION: Double Lagrange multiplier optimization of nationwide blast shelter and ABM deployments.

INPUT:

OUTPUT:

STORAGE: IDA Card Deck #107

DOCUMENTATION: IDA Study S-388, "A Study of Active and Passive
Systems for the Defense of Urban Population Centers,"
Pearsall, Grimm, Pratt (September 1971), "A Lagrange
Multiplier Method for Certain Constrained Min-Max
LANGUAGE/SYSTEM: Problems," Edward S. Pearsall, Operations Research,
Vol. 24, #1, Jan-Feb, 1976, pp. 70-91.

FTN/6400 SCOPE

COMMENTS: Source deck available. Program authors no longer at IDA.

COMPUTER PROGRAM DESCRIPTION

NAME: RESVAL

SYNOPSIS: Resource Risk Evaluation Program

TYPE: Semi-Production

USE: Given an attack and a list of resources, this program determines the overpressure level at each resource point and produces appropriate listings.

BACKGROUND: Developed in 1972 as a means of rapidly matching weapons and attack.

DESCRIPTION: An input resource file and attack file are matched to give weapon target association. Highest overpressure levels are output.

INPUT: Attack file
Resource file

OUTPUT: Reports of risk level
IDA Run #97

STORAGE: IDA Card Deck #89

DOCUMENTATION: Attached description.

LANGUAGE/SYSTEM: FTN/6400 SCOPE
FTN/3600 SCOPE?

COMMENTS: Written to be used with Emergency Broadcast System stations. Can be adapted to other resources.

For a given attack this program finds weapons within a certain distance, scaled by weapon yield, of each of a set of input resources. For each resource point it prints those weapons affecting the result. The attack used has six subattacks and results are printed for each of the subattacks.

The attack data are read first. Each weapon latitude, longitude, cosine of the latitude, and scaled critical distance are stored. To speed the search procedure the weapons are first ordered in latitude. Then the weapons are separated into five-minute strips in latitude. For each resource the weapon search is over three of these strips. This is accomplished through a dictionary which gives the range of weapon numbers, ordered in latitude, over these strips.

When a weapon association with a resource is found, these pertinent data are output. A summary gives the overall fraction of resources surviving.

```
      PROGRAM RESVAL(INPUT,OUTPUT,TPRES,TPWP)
      DIMENSION WPLAT(3300),WPLON(3300), WPDSS(3300), COSLAT(3300) ,
     1 LTA(3300),LTB(3300),LTC(3300) ,LTD(3300),LTE(3300),LTF(3300) ,
     2 IORDL(3300) ,LTPT(1500)
      CALL NOPCHKF(0)
      MP = 5LINPUT
      MQ = 6LOUTPUT
      NR = 5LTPRES     Resource Tape
      NW = 4LTPWP      attack tape
      IOEP = 0 }  input data tape type, 0 is Corps of Engineers, 1 is OEP data (A)
      IOEP = 1 }
      JCLP = 0 }  If not 1 suppress printing maximum Overpressure for each attack (B)
      JCLP = 1 }
      RADCON = 3.14159265/180.
      RADCV = RADCON/60.
      IWC = 0
      ISTCT = 0
      ISTOP = 0
      TPSID = 4.1      Inclusion distance limit for 1MT weapon (n mi.) (C)
      ICTA = 0
      ICTB = 0
      ICTC = 0
      ICTD = 0
      ICTE = 0
      ICTF = 0
      IC12 = 0
      IC123= 0
      IC14 = 0
      IC15 = 0
      IC16 = 0
      SMLLA = 99999.
      BIGLA = 0.
      BIGRAD = 0.
      LINECT = 12
      WRITE(MQ,6)
6     FORMAT( 1H1, //////,40X,* EBS STATIONS DESTROYED BY SIX OCD ATTACK
     1S* , ////)
      READ(NR,7)IZILCH
7     FORMAT(A10)
      IF(IOEP .EQ. 1) GO TO 13
      WRITE(MQ,15)
15    FORMAT( ///////, 50X, * ENGINEER DATA*)
      GOTO 14
13    CONTINUE
      WRITE (MQ,16)
16    FORMAT(///////, 58X,*OEP DATA*)
14    CONTINUE
      WRITE(MQ,72)
      WRITE(MQ,73)
      WRITE(MQ, 75)

                                                          Read attack tape
10    CONTINUE
      IWC = IWC + 1
      READ( NW,12) WLD,WLM, WLS, WLOD, WLOM, WLOS,WYLD,
     1 LTA(IWC), LTB(IWC) , LTC(IWC), LTD( IWC),LTE(IWC), LTF(IWC)
12    FORMAT( 3F3.0,F4.0,2F3.0, 6X, F6.2,23X, 6I1)
      TMP = WLD + WLM/60. + WLS/3600.
      TMA = TMP*60.
      WPLAT( IWC) = TMA
      IF(TMA .LT. SMLLA) SMLLA = TMA
      IF( TMA .GT. BIGLA) BIGLA = TMA
      TMPP = WLOD + WLOM/60. + WLOS/3600.
      TEMP = TMP*RADCON
      COSLAT(IWC ) = COS(TEMP)*60.
      WPLON(IWC) = TMPP
      DST = TPSID* WYLD**0.33333333  (C)
      IF(DST .GT. BIGRAD) BIGRAD = DST
      WPDSS(IWC) = UST*DST
      IF(LTF(IWC) .EQ. 1 .AND. LTE(IWC) .EQ.1) LTF(IWC) = 0
      IF(LTE(IWC) .EQ. 1 .AND. LTD(IWC) .EQ.1) LTE(IWC) = 0
      IF(LTD(IWC) .EQ. 1 .AND. LTC(IWC) .EQ.1) LTD(IWC) = 0
      IF(LTC(IWC) .EQ. 1 .AND. LTB(IWC) .EQ.1) LTC(IWC) = 0
      IF(LTB(IWC) .EQ. 1 .AND. LTA(IWC) .EQ.1) LTB(IWC) = 0
      IF( EOF(NW) .EQ. 0) GOTO 10                               Read Attack Tape

      CALL ORDER(WPLAT,IORDL,IWC) - Order weapons by increasing latitude
                        WPLAT(IORDL(1)) has lowest latitude, WPLAT(IORDL(IWC))
                            has highest.

      SF = SMLLA/12.          This subroutine is 6400 unique, but      Find Pointer Index
      ISF = SF                  could be replaced by NEVONS STANDARD
      IND = ISF + 1             subroutine FORD
      DO 21 J = 1,ISF
      LTPT(J) =  1
21    CONTINUE
      DO 22 J = 1,IWC
      ILK = IORDL(J)
      FAC = WPLAT(ILK)/12.
      IFAC = FAC
24    CONTINUE
      IF(IFAC.LT.IND) GO TO 22                                Weapons w/in 1/12° strip
C     LTPT(I) CONTINAES VALUE OF 1ST WPN OF LAT MORE THAN 12*I - for one value of LTPT(I)
      LTPT(IND) = J
      IND = IND + 1
      GO TO 24
22    CONTINUE                                                Zero Pointer Index

                                                              Read Two Resource
30    CONTINUE                                                  Record.
      IF(ISTOP .NE. 1, GO TO 34                               Initialize
      REWIND NR
      GO TO 100
C     AND SUMMARIZE
34    CONTINUE
      IATA =0
      IATB = 0
      IATC = 0
      IATD = 0
      IATE = 0
      IATF = 0
      DMA = 99999.
      DMB = 99999.
      DMC = 99999.
      DMD = 99999.
      DME = 99999.
      DMF = 99999.                                            Initialize

      IF( IOEP .EQ. 1) GO TO 41 (A)                           Read Data Tape
      READ(NR,31) NME1,NME2, IRSAC , ISLA, IAM, IFM             Record
     1, IUTM1,IUTM2, ISLD, ISLM,ISLS,ISLOD,ISLOM, ISLOS
31    FORMAT( 2A10,2A4, 2A4, A10,A1, 6I3)
      IF( IRSAC .NE. 4H85SA) GO TO 23
      IF(ISLA .NE. 4H0035) GO TO 23
      ISTOP = 1
23    CONTINUE
      IF( ISLD .NE. 0) GO TO 37
      WRITE(MQ,38)NME1,NME2,IAM,IFM
38    FORMAT(1H0, *FOR *,2A10, * STATION *A4,1X,A4, * NOT USED BECAUSE C
     1OORDINATES ARE MISSING*)
      GO TO 30
41    CONTINUE
      READ(NR,42) IRSAC,IAM,IFM,ISLOD,ISLOM,ISLOS,ISLD,ISLM,ISLS,
     1 NME1,NME2
42    FORMAT( 2X,A4,18X,A4,A4,9X,I3,2I2,2X, 3I2,A10,A7)
      IF(IRSAC .EQ. 4H99T2) ISTOP = 1
37    CONTINUE                                              Read Data Tape Record
      ISTCT = ISTCT + 1
      ITMP = 3600 *ISLD + 60*ISLM + ISLS                    Find Weapon
      XTMP = ITMP                                           association
      SLAT = XTMP/60.
      ITEM =  3600*ISLOD + 60*ISLOM + ISLOS
      XTEM = ITEM
      SLON = XTEM/3600.
      TEMP = SLAT*RADCV
      STCOS = COS(TEMP)*60.
      SLALO = SLAT - 12.
      SFAC = SLAT/12.     ISFAC is location in dictionary to look for weapon number - attack
      ISFAC = SFAC     -
      IST = LTPT(ISFAC + 1)   Starting point in ordered weapon list
      IWN = IST
33    CONTINUE
      IUSE = IORDL(IWN)
      IF(IUSE .GT. IWC .OR. IUSE .LT. 1) GO TO 60
      IF(WPLAT(IUSE).GT. SLALO) GO TO 32                    Try new Weapon
      IWN = IWN + 1
      GO TO 33
32    CONTINUE
      IUSE = IORDL(IWN)
      IF(IUSE .GT. IWC .OR. IUSE .LT. 1) GO TO 60
      DLAT = WPLAT(IUSE) - SLAT
      IF(DLAT.GT. BIGRAD) GOTO 60
      DLON = WPLON (IUSE)  - SLON
      IF(  ABS(DLON) .LE. 0.4) GO TO 35
      IWN = IWN + 1
      GO TO 32
35    CONTINUE
      DLON = 0.5*DLON*(COSLAT(IUSE) + STCOS)
      DSQ = DLAT*DLAT + DLON*DLON
      IF( DSQ.LE. WPDSS(IUSE) ) GO TO 36
      IWN = IWN + 1                                         Try new Weapon
      GO TO 32
36    CONTINUE

C     WPN HITS RESOURCE                                     Record for each
      DSQ = DSQ *TPSID*TPSID/WPDSS(IUSE)                    attack
      IF( LTA( IUSE) .EQ. 0) GO TO 81
      IATA =  IATA + 1
      IF( DSQ .LT. DMA )  DMA = DSQ
81    CONTINUE
      IF( LTB( IUSE) .EQ. 0) GO TO 82
      IATB =  IATB + 1
      IF( DSQ .LT. DMB )  DMB = DSQ
```

```
82    CONTINUE
      IF( LTC( IUSE) .EQ. 0) GO TO 83
      IATC =  IATC + 1
      IF( DSQ .LT. DMC )  DMC = DSQ
83    CONTINUE
      IF( LTD( IUSE) .EQ. 0) GO TO 84
      IATD =  IATD + 1
      IF( DSQ .LT. DMD )  DMD = DSQ
84    CONTINUE
      IF( LTE( IUSE) .EQ. 0) GO TO 85
      IATE =  IATE + 1
      IF( DSQ .LT. DME )  DME = DSQ
85    CONTINUE
      IF( LTF( IUSE) .EQ. 0) GO TO 86
      IATF =  IATF + 1
      IF( DSQ .LT. DMF )  DMF = DSQ
86    CONTINUE
      IWN = IWN + 1
      GO TO 32


60    CONTINUE
      PRESA = 0.
      PRESB = 0.
      PRESC = 0.
      PRESD = 0.
      PRESE = 0.
      PRESF = 0.
      IF(IATA .NE. 0) ICTA = ICTA + 1
      IF(IATB .NE.0) ICTB = ICTB + 1
      IF( IATC .NE.0) ICTC = ICTC + 1
      IF(IATD .NE. 0) ICTD = ICTD + 1
      IF(IATE .NE.0) ICTE = ICTE + 1
      IF( IATF .NE.0) ICTF = ICTF + 1
      IF(JCLP .NE. 1) GO TO 87
      IF (IATA .EQ. 0) GO TO 91
      DCALL = DMA
      DCALL = SQRT(DCALL)
      CALL CLPRES( DCALL,PRESA)
91    CONTINUE
      IF (IATB .EQ. 0) GO TO 92
      DCALL = DMB
      DCALL = SQRT(DCALL)
      CALL CLPRES( DCALL,PRESB)
92    CONTINUE
      IF (IATC .EQ. 0) GO TO 93
      DCALL = DMC
      DCALL = SQRT(DCALL)
      CALL CLPRES( DCALL,PRESC)
93    CONTINUE
      IF (IATD .EQ. 0) GO TO 94
      DCALL = DMD
      DCALL = SQRT(DCALL)
      CALL CLPRES( DCALL,PRESD)
94    CONTINUE
      IF (IATE .EQ. 0) GO TO 95
      DCALL = DME
      DCALL = SQRT(DCALL)
      CALL CLPRES( DCALL,PRESE)
95    CONTINUE
```

```
      IF (IATF .EQ. 0) GO TO 96
      DCALL = DMF
      DCALL = SQRT(DCALL)
      CALL CLPRES( DCALL,PRESF)
96    CONTINUE
87    CONTINUE
      IF(LINECT .LE. 54) GO TO 71
      WRITE(MQ,72)
72    FORMAT(1H1,//, 1H ,5X,9HCITY NAME ,6X, 1X, 4HRSAC, 1X, 4H SLA ,
     1 2X, 4H AM  ,1X, 4H FM  ,1X  11H    UTM      ,2X,
     2  9HLATITUDE  ,2X, 9HLONGITUDE ,6X,
     3  42HATTACK RESULTS ----=KILLED  0=NOT KILLED       )
      WRITE(MQ,73)
73    FORMAT( 1H  ,21X, 4H NO.  ,1X, 4H NO.  2X, 4HCALL,1X, 4HCALL,
     1 1X,11HCOORDINATES  ,2X, 9H D  M  S   ,2X, 9H D  M  S      ,
     2 6X, 45HATT.1  ATT.2  ATT.3  ATT.4  ATT.5  ATT.6            )
      WRITE(MQ, 75)
75    FORMAT( 1H , 32X, 4H NO. , 1X, 4H NO.  ,/)
      LINECT = 6
71    CONTINUE
      IF(IOEP .EQ. 1) GO TO 76 (A)
      WRITE(MQ,74) NME1,NME2,IRSAC,ISLA,IAM, IFM,  IUTM1,IUTM2 ,
     1  ISLD,ISLM, ISLS, ISLOD, ISLOM, ISLOS,IATA, IATB,IATC,IATD,IATE,
     2 IATF
74    FORMAT( 1H , 2A10,1X, A4,1X, A4,2X, A4,  1X, A4, 1X,
     1 A10, A1, 2X, 3I3, 2X, 3I3, 6X, 6(I5,2X))
      GO TO 77
76    CONTINUE
      WRITE(MQ,78) NME1 ,NME2,IRSAC, IAM,IFM, ISLD, ISLM,ISLS,ISLOD,
     1 ISLOM, ISLOS,IATA,IATB,IATC, IATD,IATE,IATF
78    FORMAT( 1H ,A10,A7,4X,A4,  5X,2X,A4,1X,A4,1X,  11X,2X,3I3,
     1  2X, 3I3, 6X, 6(I5,2X))
77    CONTINUE
      IF(JCLP .NE. 1) GO TO 89(B)
      IF( IATA .EQ. 0 .AND . IATB .EQ.0 .AND. IATC .EQ.0 .AND.IATD
     1 .EQ. 0 .AND . IATE .EQ. 0 .AND. IATF.EQ.0) GO TO 97
      WRITE(MQ,79) PRESA,PRESB,PRESC,PRESD,PRESE,PRESF
79    FORMAT( 1H , 56X, 23HMAXIMUM OVERPRESSURE = , 6F7.1)
        GO TO 98
97    CONTINUE
      WRITE (MQ, 99)
99    FORMAT(1H )
98    CONTINUE
      LINECT = LINECT + 1
89    CONTINUE
      LINECT = LINECT + 1

      IF( IATA .NE.0 .OR. IATB .NE.0) IC12  = IC12 +1
      IF( IATA. NE.0 .OR. IATB.NE.0 .OR.IATC.NE.0) IC123 = IC123 +1
      IF( IATA. NE. 0 .OR. IATB .NE.0.OR. IATC.NE.0 .OR. IATD.NE.0)
     1 IC14 = IC14 + 1
      IF( IATA. NE. 0 .OR. IATB .NE. 0 .OR. IATC .NE. 0 .OR.IATD .NE.0
     1 .OR. IATE. NE.0) IC15  = IC15 + 1
      IF( IATA.NE. 0 .OR. IATB .NE. 0 .OR. IATC .NE . 0.OR.IATD.NE.0
     1 .OR.IATE .NE. 0 .OR. IATF. NE. 0) IC16 = IC16 +1
      GO TO 30


100   CONTINUE
      WRITE (MQ, 101) ISTCT
```

Record for each attack

Output this weapon record

Output the weapon associated

Add to summary data

Add to summary data

Read new resource record.

Read attack tape

Output summary

```
101   FORMAT(1H1,//// ,* NATIONWIDE SUMMARY ON*,I5,* RESOURCE POINTS*,
     1  /////)
      WRITE(MO,102) ICTA,ICTB,ICTC,ICTD, ICTE,ICTF
102   FORMAT(  1H0,     *POINTS DESTROYED BY ATTACK 1 =*  ,I5,/ ,
     1            1H0,     *POINTS DESTROYED BY ATTACK 2 =*  ,I5,/ ,
     2            1H0,     *POINTS DESTROYED BY ATTACK 3 =*  ,I5,/ ,
     3            1H0,     *POINTS DESTROYED BY ATTACK 4 =*  ,I5,/ ,
     4            1H0,     *POINTS DESTROYED BY ATTACK 5 =*  ,I5,/ ,
     5            1H0,     *POINTS DESTROYED BY ATTACK 6 =*  ,I5,/ )
      WRITE(MO,103) IC12,IC123, IC14,IC15,IC16
103   FORMAT( ///,1H0, *POINTS DESTROYED BY ATTACKS 1 OR 2 =*,I5,/,
     1               1H0, *POINTS DESTROYED BY ATTACKS 1 OR 2 OR 3 =*
     2  ,I5,/,      1H0, *POINTS DESTROYED BY ATTACKS 1 OR 2 OR 3 OR 4 =*
     3  ,I5,/,      1H0, *POINTS DESTROYED BY ATTACKS 1 OR 2 OR 3 OR 4 OR
     45 =*
     5  ,I5,/,      1H0, *POINTS DESTROYED BY ATTACKS 1 OR 2 OR 3 OR 4 OR
     65 OR 6 =*  ,I5)
      STOP 6400
      END
      SUBROUTINE CLPRES(DIST,PRES)
C   COMPUTES PRESSURE GIVEN DISTANCE FOROA 1 MT SURFACE BURST
C    FIT TO VALUE IN ENW
      IF( DIST .GT. 0.0001) GO  TO 4
      DIST = 0.0001
4     CONTINUE
      XLDIST = ALOG10(DIST)
      IF (DIST .GE. 1.147) GO TO 2
        SLOPE = -2.255
      GO TO 3
2     CONTINUE
       SLOPE = -1.825
3     CONTINUE
      XLPRES = 1.3010 + SLOPE*(XLDIST - 0.0596)
      PRES = 10.**XLPRES
      IF(PRES .GT. 9999.) PRES = 9999.
      RETURN
      END
          IDENT      ORDER
         ENTRY ORDER
 BYTE     EQU 10
 ACRES    EQU 1024
*BYTE    EQU 6         FOR ORDER6
*ACRES   EQU 64
         USE TALLY
TLLY     VFD 40/TALLY
TALLY    BSSZ ACRES
         USE 0
          VFD         30D/SLORDER,300/3D
ORDER    PS
          SX6         A0
          SA6         =SEXSV
          SA1         A1
          SB1         X1
          SA1         A1+1
          SB2         X1
          SA1         A1+1
          SB3         X1
          SB4         ONE
* SETTING B4 TO ADDR OF 0 WORD WOULD DO ORDER WITHOUT REGARD TO SIGN
NOSINE    SX6         B1
         SA6 =SBA
```

Output Sum.

Pressure vs. distance
Subroutin

Ordering Subroution
6400 [illegible]
Replace by FORD
if struct FORTRAN
calculator

```
          SX7 B2
          SA7 =SBB
           SX7        B3
           SA7        =SBC
          SA3 B3
           SB5        X3-1
           NE         B5,MTONE
           SX6        1
           SA6        B2
           SA2        =SEXSV
           SA0        X2
           JP         ORDER
MTONE      SB6        X3                    KEEP LENGTH IN B6
          MX0 60-BYTE
          SA0 B2        A0 IS INDICES
          SB2 B1      B1 IS[DATA]
          SB1 1
           LT         B6,B1,ORDERX          EXIT IF NN LE TO ZERO
          SB5 -B1
          SB3 TALLY
          MX5 1       READY PLOOP
          SB7 B0
          SX3 B1
          SA2 B4
          SX2 X2+B5
*         ZR X5,*+1
          MX5 0
*         SX6 B5+B2
* NOW PRESET INDEX TABLE AND FIX UP DATA
PLOOP     SA1 B7+B2    FETCH DATA ITEM
          IX6 X6+X3
          SA6 B7+A0    STORE DATA ADDRESS IN INDEX
          BX7 X1-X5    ON OPTION COMPL. LEFT BIT
          SA7 A1        SO NEGATIVE IS > POSITIVE
          SB7 B7+B1
          LT B7,B6,PLOOP
          BX7 X5
          SA7 =SCOMPSW
          SB4 B0       PRESET SHIFT COUNT
* THROUGH PRELIMINARIES-INDEX WORK VIA B4(SHIFT)
MAIN      SB7 B0
          SX7 B1        FOR FREQUENCY COUNTING
CTLOOP    SA1 B7+B2     GET A DATA ITEM
          SB7 B7+B1
          AX5 B4,X1
          BX3 -X0*X5
          SA4 X3+B3
          IX6 X7+X4
          SA6 A4
          LT B7,B6,CTLOOP
*  CUMULATE FREQUENCY COUNT
          SB7 -TALLY-ACRES+1
          SA1 TALLY
          SX6 X1+B5
          SA6 A1
NSLOOP    SA1 A1+B1
          IX6 X6+X1
          SX7 B7+A1
          SA6 A1
          NZ X7,NSLOOP
*  NOW SORT
```

```
         SB7 B6-B1
         SA5 B2          JUST TO HIDE B2
         SB2 A0        B2 AND A0 ARE BOTH [INDEX]
SOLOOP   SA1 B7+A0     FETCH INDICES FROM BOTTOM
         SB7 B7-B1     BACK POINTER
         SA2 X1        FETCH DATA
         SX7 A2        SAVE DATA ADDRESS
         AX5 X2,B4     SHIFT DATA
         BX3 -X0*X5      AND GET CURRENT BYTE
         SX2 B7        FOR INDEXING PURPOSES
         SA4 X3+B3     FETCH TALLY OF BYTE
         LX7 30
         SX6 X4+B5     BACK OFF TALLY
         SA3 X4+B2     GET INDEX+TALLY+
         IX7 X7+X3     PUT IN THIS DATA ADDRESS
         SA7 A3        PUTAWAY
         SA6 A4
         PL X2,SOLOOP
         SB2 A5
         SB5 30
*  NOW TIDY UP FOR NEXT PASS
         SB7 B0
XLOOP    SA1 B7+A0     GET INDEX
         AX7 X1,B5       RIGHT 30
         SA7 A1
XL       SB7 B7+B1
         LT B7,B6,XLOOP
         MX6 0
         SX4 ACRES
         SX7 B3
         SX5 B1+B1
         SA7 B3-B1
         BX7 X0-X0
XLL      SA6 A7+B1     CLEAN OUT COUNT
         IX4 X4-X5
         SA7 A6+B1
         NZ X4,XLL
*    MORE REPAIRS FOR NEXT PASS
         SB5 -B1
         SB4 B4+BYTE
         SX1 B4-60
         NG X1,MAIN
*  THROUGH--NOW CLEAN UP AND EXIT
         SA5 COMPSW
         SB4 30
         SX4 B2+B5
         BX0 X2
         SB7 B6+B5
WLOOP    SA1 B7+B2
         BX6 X1-X0
         SA3 B7+A0
         IX7 X3-X4
         SB7 B7+B5
         SA7 A3
         SA6 A1
WL       PL B7,QLOUP
          SA1        =SBA
          SB1        X1
          SA1        =SBB
          SB2        X1
          SA1        =SBC
```

9

```
          SB3       X1
          SA2       =SEXSV
          SA0       X2
         EQ ORDER
          ENTRY     ORDNS
          VFD       30D/SLORDNS,30D/3
ORDNS     DATA      0
          SA1       ORDNS                GET RETURN ADDRESS
          BX6       X1                   TRANSFER TO MAIN ROUTINE EXIT
          SA6       ORDER
          SB4       ZERO
          JP        NOSINE
ORDERX    SB1       LE0
          RJ        =XABRTJOB
ZERO      DATA      0
ONE       DATA      1
LE0       DIS       ,*ORDER-NN LE TO 0*
         END
```

COMPUTER PROGRAM DESCRIPTION

NAME: ADAGIO

SYNOPSIS: Nationwide Evacuation Analysis

TYPE: Production

USE: To study nationwide evacuation under a variety of assumptions.

BACKGROUND: Developed over a number of years to study nationwide evacuation. Used as a starting point for Crisis Relocation Planning.

DESCRIPTION: Basic relocation of population from risk areas to reception areas to minimize travel distance. A variety of options and types of printouts available.

INPUT: Blast Risk data, Fallout Risk data, Population data, Economic data

OUTPUT: IDA printouts 24-26, 30-36, 45-48, 80, 105, 113 relocated population data files

STORAGE: see below

DOCUMENTATION: IDA Paper P-702 "A Study of National Travel Requirements for Strategic Evacuation, Leo A. Schmidt, March 1970; IDA Paper P-1067, "The Use of the ADAGIO Computer Program in Strategic Evacuation Analysis," Leo A. Schmidt, October 1974; IDA Paper P-1183, "Interactive ADAGIO Computer Program as an Aid to Crisis Relocation Planning," Leo A. Schmidt, November 1975

LANGUAGE/SYSTEM: see below

NAME	STORAGE	LANGUAGE	VARIATION
SCATTER4	IDA Card Deck 90	FTN/1604	Original Evacuation Study Program-1966
ADAGIO-RSAC	IDA Card Deck 75	FTN/6400SCOPE	Final Version with 1960 Census in RSAC Format
DAGIO-74	IDA Card Deck 85	FTN/6400SCOPE	IDA Version of CRP Program
ADAGIO-CRP	DCPACC	FTN/3600SCOPE	DCPA Version of CRP Program
DAGIO-NEPA	DCPACC	FTN/3600SCOPE	DCPA Version of Externally Defined Risk Program
ADAGIO-75	IDA Card Deck 202	FTN/6400SCOP	Current Batch ADAGIO Program
DAGIO-TERM	CDC KRONOS Tape TKW2212	FTN/KRONOS BATCH	Current Interactive ADAGIO Program

Preceding Page Blank - NOT FILMed

COMPUTER PROGRAM DESCRIPTION

NAME: TSTGD

SYNOPSIS: Contour Printer Plot Routine

TYPE: Development

USE: To obtain printer plots of values of 2-dimensional functions with shading proportional to function value.

BACKGROUND: Developed to display population densities, fallout fields, etc., using only normal printer facilities.

DESCRIPTION: A function of 2 variables is smoothed, based on several smoothing options. Plots can be multipaged with provisions for margin, fiducial marks, etc., automatically provided by the program.

INPUT:
Data files, parameter cards

OUTPUT: IDA runs 68, 108--plots set of population densities in urbanized areas available

STORAGE: IDA Card Decks#59, 46

DOCUMENTATION: Memo Eli Williams to L. Schmidt dated 17 April 1973, entitled: Subroutine Write-Up

LANGUAGE/SYSTEM: FTN/6400 SCOPE

COMMENTS: Copy of source program and documentation furnished to DCPACC in 1976

COMPUTER PROGRAM DESCRIPTION

NAME: JLAM

SYNOPSIS: Compute Analytic Sums

TYPE: Single Use

USE: Computes values of sums for use in analysis

BACKGROUND:

DESCRIPTION: Computes and prints sums $\sum_{i=1}^{N} \frac{1}{i}$, $\sum_{i=1}^{N} \frac{1}{i^{1/2}}$, $\sum_{i=1}^{N} \ln i^{1/2} / i^{1/2}$ $\sum_{i=1}^{N} \ln^{2} i^{1/2} / i^{1/2}$

INPUT: IN PROGRAM

OUTPUT:

STORAGE: IDA Card Decks #81, 82

DOCUMENTATION: IDA Paper P-870, "Analytic Models of Nationwide Urban Fatalities from a Nuclear Attack," Leo A. Schmidt, July 1972

LANGUAGE/SYSTEM: RUN(FORTRAN)/6400 SCOPE

COMMENTS:

COMPUTER PROGRAM DESCRIPTION

NAME: TEST

SYNOPSIS: Fire Spread Model

TYPE: Development

USE: To study fire spread in urban tracts

BACKGROUND: Developed by F. Miercort as a synopsis of the IITRI Fire Spread Model. Faster running with multiple weapon inputs. Written to be compatible with TELOS program as it existed in 1973.

DESCRIPTION: Fire spread by radiation and firebrands is modeled. Detailed bookkeeping by building type and tract of blast damage and fire histories, ignitions imposed externally. Main coding is in subroutine "FIRE" with main program a temporary driver

INPUT:

Data cards

OUTPUT: Printouts

STORAGE: IDA Card Deck 131, 217

DOCUMENTATION: IDA Paper P-988, "Description of a Fast-Running Fire-Spread Model," Frederic A. Miercort, December 1973

LANGUAGE/SYSTEM: FTN/6400 SCOPE

COMMENTS: Paper contains detailed program description.

COMPUTER PROGRAM DESCRIPTION

NAME: POPPOP

SYNOPSIS: Probabilistic Fire Spread

TYPE: Development

USE: Fire Interactions Analyses

BACKGROUND: Developed as part of Blast Fire Interaction Study as an experimental program to assess various probabilistic effects

DESCRIPTION: In a Monte Carlo simulation, an initial set of ignitions are made at interactions of an m x n grid. These ignited points can propagate fire in each of the 4 cardinal directions each time period with a certain probability. The resulting fire pattern is printed.

INPUT: Parameters in program.

OUTPUT: Printouts

STORAGE: IDA Card Deck #204

DOCUMENTATION: Forthcoming paper; attached description

LANGUAGE/SYSTEM: RUN(FORTRAN)/6400 SCOPE

COMMENTS:

This program is a simulation of probabilistic ignition possibilities during a series of time periods, ICNT. In this program the array ISTATE(I,J) indicates whether a building located at I,J in a grid is burning; ISTATE = 0 means no burning, -1, a burning was just ignited, and +n, the building has been burning for n periods. An unignited building may be ignited if one of its neighbors (at I-1,J; I+1,J; I,J-1, I,J+1) has been burning for NBRN periods or less. This is tested in the section called Search Matrix, in the two DO loops ending at statements 111 and 112. For each neighbor which may ignite a building a random number is drawn and compared with PGO or PVARAR(I,J) to see if the structure is ignited.

In the section called Update Matrix the DO loop terminating in statement 201 tests if any buildings may still ignite others. If so, the matrix ISTATE is updated in the DO loop terminating in statement 202. the time is incremented, and another search is made; if not, this trial is terminated.

There are NTRY trials. Each time a building at I,J is ignited the array ISUM(I,J) is incremented by one as a record of burnings over all the trials.

The contents of ISTATE, ISUM, or PVARAR may be printed by the various report sections. The remainder of the program is initialization or control activities.

```
      PROGRAM POPPOP(INPUT,OUTPUT)
C
C
C
      DIMENSION ISTATE(60,60),ISUM(60,60)
      DIMENSION PVARAR(60,60)
      DIMENSION PGOAR(20),PSTAR(20)
C                                                                   [ Set Parameters
C
      IPALL = 0    } Print results after Each Trial (A)
      IPALL = 1
      IPPTH = 1    } Print Results after Each Time Step (B)
      IPPTH = 0
      NTRYP = 200  )
      NTRYP = 10   } number of Trials To Print (C)
      NTRYP = 5    }
      NTRYP = 25   )
      NBRN = 3     }
      NBRN = 10    } number of Burning Periods (D)
      NBRN = 1     }
      XNBRN = NBRN
      IPVAR = 0    } constant Transition Probability (E)
      IPVAR = 1
C***********************************//////////////////////////////
      PVSIG = 0.1        Range of Variability (F)
      NITPV = 10
      DO 22 I = 1,20
      PGOAR(I) = 0.
      PSTAR(I) = 0.
      PSTAR(I) = 0.01  Initial Ignition Probability (G)
22    CONTINUE
      PGOAR(1) = 0.5   Transition Probability (H)
      MP = 5LINPUT
      MQ = 6LOUTPUT
      SEED = 3.14159265   )
      SEED = 2.7536       }
      SEED = 532.56238    }
      SEED = 9582.5520    }
      SEED = 235.45230    }
      SEED = 58.6665      }
      SEED = 253.444      }  Random number Generator
      SEED = 3.456872     }  Initialization
      SEED = 147.665      }
      SEED = 523.44578    }
      ZILCH = RANF(SEED)  )
      A1 = 24 - 1
      A3 = 24 - 3
      A5 = 24 - 5
      A7 = 24 - 7
      A9 = 24 - 9
      A10 = 24 - 10
      NTRY = 10    }
      NTRY = 1000  } number of Repetitions (I)
      NTRY = 100   }
      NRI = 23
      NRJ = 23     } matrix Size (J)                                    Set Parameters
C                                                                     __|
C
      ICASE = 0                                                       Set new Transition
40    CONTINUE                                                            Probability
      ICASE = ICASE + 1
      PGO = PGOAR(ICASE)
      PST = PSTAR(ICASE)
      IF(PGO .EQ. 0.) GO TO 5000
      WRITE(MO,42) ICASE,PGO ,NTRY ,NBRN
42     FORMAT(1H1, ////,20X,* *******RESULTS FOR CASE NO. *,I4,* *******
     1*******,////,20X,*   THYE VALUE OF TRANSITION PROBABILITY IS *,
     2  ,F8.4, *  NUMBER OF TRIALS =*,I5,* BURNING DURATION IS*,I5,//)
      AL = (1.-(1. - PGO)**(1./XNBRN))/PGO          D
      PGO = AL*PGO
      DO 21 I = 1,NRI
      DO 21 J = 1,NRJ
      ISUM(I,J) = 0
21    CONTINUE
      ISUMA = 0
      ISUMB = 0
      ISUMC = 0
      ISUMD = 0
      ISUME = 0
      ISUMF = 0
      IT = 0                                                  Start new Trial
C
C
100   CONTINUE
      IT = IT + 1
C
      IF(IPVAR .NE. 1) GO TO 303   E
      ITEM = IT/NITPV
      ITM = IT - ITEM*NITPV -1
      IF(ITM .NE. 0) GO TO 303
      DO 302 I = 1,NRI
      DO 302 J = 1,NRJ
      PVARAR(I,J) = PGO + 2.*PVSIG*(RANF(0) - 0.5)   F
302   CONTINUE
      IF( IT .GT. 30) GO TO 308
      WRITE(MO,304)
304   FORMAT( 1H1, /////,20X,* ((((((((TRANSITION PROB FOR NEXT RUNS))))
     1))))))*,////)
      WRITE(MO,209) (J,J = 1,NRJ)
      DO 306 I = 1,NRI
      WRITE(MO,305) I,(PVARAR(I,J),J=1,NRJ)
305   FORMAT( 1H0, * I =*,I5,5X, 23F5.3,//)
306   CONTINUE
308   CONTINUE
303   CONTINUE                                          Initial Ignition
C
      DO 101 I = 1,NRI   J
      DO 101 J = 1,NRJ
      ISTATE (I,J) = 0
101   CONTINUE             ~ {IF( PSTAR(I).GT. RANF(0)) GO TO 101   G
      ISTATE(12,12) = 1       { ISTATE(I,J)=1
      ICNT = 0                                           Initial Ignitions
105   CONTINUE
      ICNT = ICNT + 1
      DO 111 I = 1,NRI                      Search Matrix  Start new Time Period
      DO 112 J = 1,NRJ
      IF(I .EQ.1 .OR. I.EQ. NRI)GO TO 123
      IF(J .EQ.1 .OR. J.EQ. NRJ)GO TO 124
      IF(ISTATE(I,J) .NE.0) GO TO 150
      DO 141 KK = 1,4
      IF(KK .NE. 1) GO TO 142
      IF(ISTATE(I-1,J) .LE. 0) GO TO 147
      IF(ISTATE(I-1,J) .GT. NBRN) GO TO 147 (D)
      GO TO 146
142   CONTINUE
      IF(KK .NE. 2) GO TO 143
      IF(ISTATE(I+1,J) .LE. 0) GO TO 147
      IF(ISTATE(I+1,J) .GT. NBRN) GO TO 147 (D)
      GO TO 146
143   CONTINUE
      IF(KK .NE. 3) GO TO 144
      IF(ISTATE(I,J-1) .LE. 0) GO TO 147
      IF(ISTATE(I,J-1) .GT. NBRN) GO TO 147 (A)
      GO TO 146
144   CONTINUE
      IF(KK .NE. 4) GO TO 145
      IF(ISTATE(I,J+1) .LE. 0) GO TO 147
      IF(ISTATE(I,J+1) .GT. NBRN) GO TO 147 (C)
      GO TO 146
145   CONTINUE
146   CONTINUE
      IF(IPVAR .EQ.1) GO TO 148  (E)
      IF(RANF(0) .GT. PGO) GO TO 141  (H)
      ISTATE(I,J) = -1
      GO TO 150
148   CONTINUE
      IF(RANF(0) .GT. PVARAR(I,J))GO TO 141  (E)
      ISTATE(I,J) = -1                          -- This point ignited
      GO TO 150
147   CONTINUE
141   CONTINUE
      GO TO 150
150   CONTINUE
C
123   CONTINUE
124   CONTINUE
112   CONTINUE
111   CONTINUE                                Search Matrix
C
C
      IF(IPPTH .NE.1) GO TO 206 (B)
      IF(ICNT .GT. 1) GO TO 206
      WRITE(MQ,207)ICNT
207   FORMAT(1H1,///,*/////RESULTS AFTER STEP NO*,I5,*/////*,//)
      WRITE(MQ,209) (J,J = 1,NRJ)
      DO 205 I = 1,NRI
      WRITE(MQ,211) I,(ISTATE(I,J),J = 1,NRJ)
205   CONTINUE
206   CONTINUE
C                                                         Update matrix
C
      IDONE = 1
      DO 201 I = 1,NRI
      DO 201 J = 1,NRJ
      IF(ISTATE(I,J) .EQ.-1) IDONE = 0
      IF(ISTATE(I,J) .GT. 0 .AND. ISTATE(I,J) .LT.NBRN ) IDONE = 0  (D)
201   CONTINUE
      IF(IDONE .EQ.1) GO TO 223
      DO 202 I = 1,NRI
      DO 202 J = 1,NRJ
      IF(ISTATE(I,J) .LE.0) GO TO 203
      ISTATE(I,J) = ISTATE(I,J) + 1
203   CONTINUE
```

```
      IF(ISTATE(I,J) .EQ.-1) ISTATF(I,J) = 1
 202  CONTINUE                                                    Update matrix 4
      GO TO 105
 223  CONTINUE                                                    Start New Time Period
C
C
C
      DO 221 I = 1,NRI
      DO 221 J = 1,NRJ
      IF(ISTATE(I,J).EQ.0) GO TO 221
      ISUM(I,J) = ISUM(I,J) + 1
      IF(I .LT.  1.OR. I .GT.   A1) GO TO 251
      IF(J .LT.  1.OR. J .GT.   A1) GO TO 251
      ISUMA = ISUMA + 1
      IF(I .LT.  3.OR. I .GT.   A3) GO TO 251
      IF(J .LT.  3.OR. J .GT.   A3) GO TO 251
      ISUMB = ISUMB + 1
      IF(I .LT.  5.OR. I .GT.   A5) GO TO 251
      IF(J .LT.  5.OR. J .GT.   A5) GO TO 251
      ISUMC = ISUMC + 1
      IF(I .LT.  7.OR. I .GT.   A7) GO TO 251
      IF(J .LT.  7.OR. J .GT.   A7) GO TO 251
      ISUMD = ISUMD + 1
      IF(I .LT.  9.OR. I .GT.   A9) GO TO 251
      IF(J .LT.  9.OR. J .GT.   A9) GO TO 251
      ISUME = ISUME + 1
      IF(I .LT. 10.OR. I .GT.  A10) GO TO 251
      IF(J .LT. 10.OR. J .GT.  A10) GO TO 251
      ISUMF = ISUMF + 1
 251  CONTINUE
 221  CONTINUE
C
      IF(IPALL .NE.1) GO TO  215     (A)
      IF(IT .GT. NTRYP) GO TO  215   (C)
      WRITE(MO,208) IT      ,ICNT ,PGO
 208  FORMAT(1H1, ///////,20X,*=========RESULTS FOR TRIAL NO.*,I5,*=====
     1=======*,/,*   RESULTS AFTER*,I6,* STEPS*,  *  TRANSITION PROB. =
     2*, F10.4)
      WRITE(MO,209) (J,J = 1,NRJ)
 209  FORMAT(1H0, 4X, * J = *    ,5X, 23I5,//)
      DO 210 I = 1,NRI
      WRITE(MO,211) I,(ISTATE(I,J),J = 1,NRJ)
 211  FORMAT( 1H0, * I =*,I5,5X, 23I5)
 210  CONTINUE
 215  CONTINUE
      IF(IT .LT. NTRY ) GO TO 100 (I)                             Set New Trial
C
C
      WRITE(MO,231) IT  ,PGO ,PST
 231  FORMAT( 1H1,///, 20X,*----------SUMMARRY RESULTS FROM*,I5,*TRIALS
     1--------------*,/,*      TRANS. PROB. =*,F10.4,*  IGNITION PROB. =
     2 *,F10.4,//)
      WRITE(MO,209) (J,J = 1,NRJ)
      DO 232 I = 1,NRI
      WRITE(MO,211) I,(  ISUM(I,J),J = 1,NRJ)
 232  CONTINUE
      TMP = ISUMA
      TMA = TMP/529
      TMP = ISUMB
      TMB = TMP/361
      TMP = ISUMC
      TMC = TMP/225
      TMP = ISUMD
      TMD = TMP/121
      TMP = ISUME
      TME = TMP/49
      TMP = ISUMF
      TMF = TMP/25
252   FORMAT(1H0 ,* SUMS A-B-C-D-E-F = *,6I10)
      WRITE(MQ,252)ISUMA,ISUMB,ISUMC,ISUMD,ISUME,ISUMF
      WRITE(MQ,253) TMA,TMB,TMC,TMD,TME,TMF
253   FORMAT(1H0, *  FRACTIONS A-B-C-D-E-F = *,6F10.4)
      GO TO 40
C
C
5000  CONTINUE
      STOP 6400
      END
```

Set Transiti
Probabilit

COMPUTER PROGRAM DESCRIPTION

NAME: FIRESM

SYNOPSIS: Probabilistic Fire Spread

TYPE: Development

USE: Fire Interaction Analysis

BACKGROUND: Developed from program POPPOP to have a simulation closer in nature to the IITRI fire model

DESCRIPTION: The basic probabilistic model of program POPPOP is preserved. A building burning time is determined probabilistically. Fire spread by firebrands is modeled as well as by radiation.

INPUT: Parameters in program.

OUTPUT: Printouts

STORAGE: IDA Card Deck #205

DOCUMENTATION: Forthcoming paper; attached description

LANGUAGE/SYSTEM: Run(FORTRAN)/6400 SCOPE

COMMENTS:

Preceding Page BLANK - NOT FILMED

This program is developed from program POPPOP but has random burning times and firebrands. The array TSTATE(I,J) gives the time when the building can transmit ignitions if it is already ignited; if TSTATE(I,J) is zero the building has not yet been ignited. The search procedure is somewhat different than in POPPOP. If the current time period is when a building can transmit burning then the section starting at statement 400 is entered. Each of the four neighbors to the building at I,J is tested to see if it is burning. If a neighbor has not been ignited, then random numbers are drawn to see if it will ignite by radiation. For firebrands a possible firebrand location is found. It is found using the downwind distance probabilities in arrays DISTAR and DELAR. The crosswind distribution is uniform within the limiting angle PHI. The downwind direction is always assumed to the right, i.e., in the direction of increasing I. From the nearest grid point the firebrand susceptibility factor is found. This and the firebrand output factor from the donor building and a random number are used to determine if the structure is ignited.

The reports generated are similar to those in the program POPPOP.

```
      PROGRAM FIRESM(INPUT,OUTPUT)
C
C
C
      DIMENSION ISTATE(40,40),ISUM(40,40),TSTATE(40,40)
      DIMENSION RPAR(40,40),SPAR(40,40),FPAR(40,40), SPFAR(40,40)
      DIMENSION DISTAR(10),DELAR(10)
      DIMENSION PGOAR(20),PSTAR(20)
      DIMENSION ING(40,40),JNG(40,40)
      DATA DISTAR /65.,105.,  137.,  165.,190.,220.,255.,300.,378.,483./
      DATA DELAR / 80.,40.,35.,30.,22.,30.,45.,55.,42.,60./
C
C                                                            Set Parameters
      IPALL = 0} Print Results after Each Trial Ⓐ
      IPALL = 1}
      IPPTH = 0} Print Results of Time steps Ⓑ
      IPPTH = 1}
      IPTIG = 1} Print Time to ignite matrix Ⓒ
      IPTIG = 0}
      NPPTH = 100  }   maximum time step for printing Ⓓ
      NPPTH = 200  }
      NPPTH = 1000 }
      NPPTH = 10   }
      NPPS = 1  }
      NPPS = 10 }   Print each NPPS time steps Ⓔ
      NPPS = 8  }
      NPPS = 40 }
      NTRYP = 200}
      NTRYP = 5  } number of Trials to Print Ⓕ
      NTRYP = 10 }
      NBRN = 3      }
      NBRN = 10     }
      NBRN = 1      } not used
      XNBRN = NBRN  }
      IPVAR = 0  }
      IPVAR = 1  }
      PVSIG = 0.3}  change susceptibility matrix each NITPV trials Ⓖ
      NITPV = 10
C
      TDMN = 0.5  Mean Time for Radiant Spread, hrs. Ⓗ
      BDMN = 200. not used
      PHI = 70.                               }
      TNPH = TAN(PHI*3.14159265/360.)         } angle of firebrand spread Ⓘ
C     ONLY WANT HALF THE TOTAL ANGLE
      DELX = 50.  mean distance between Grid Points Ⓙ
      DSTIG = 5.*DELX  Distance for initial ignitions (JJ)
      DELT = 0.25  Time increment, hrs Ⓚ
      WIND = 8.} Wind Speed, mph Ⓛ
      WIND = 4.}
      PBI = 0.2   }
      PBI = 0.05  }
      PBI = 0.02  } Expected firebrand ignitions from a burning building Ⓜ
      PBI = 0.    }
      PBI = 0.1   }
      PBI = PBI*WIND/4.
      RPSG = 1.   }
      SPSG = 1.   }
      FPSG = 1.   }
      SPFSG = 1.  }
      RPSG = 0.3  }
      SPSG = 0.3  }
      FPSG = 0.3      }
      SPFSG = 0.3     }  Spread of Susceptibility and Radiation Factor Ⓡ
      RPSG = 5.       }
      SPSG = 5.       }
      FPSG = 5.       }
      SPFSG = 5.      }
C
      DO 22 I = 1,20    }
      PGOAR(I) = 0.     }
      PSTAR(I) = 0.     }
      PSTAR(I) = 0.01   }  Initial Ignition and Transmission Probabilities Ⓖ
22    CONTINUE          }
      PGOAR(1) = 0.5    }
      PSTAR(1) = 0.8    }
      PGOAR(2) = 0.4    }
      PSTAR(2) = 0.5    }
      MP = 5LINPUT
      MQ = 6LOUTPUT
      SEED = 3.14159265   }
      SEED = 2.7536       }
      SEED = 532.56238    }
      SEED = 58.6665      }
      SEED = 6.45723      }
      SEED = 253.444      }  Random Number Generator
      SEED = 235.45230    }  Initialization
      SEED = 9587.5520    }
      SEED = 8.421307     }
      ZILCH = RANF(SEED)  }
      A1 = 24 - 1
      A3 = 24 - 3
      A5 = 24 - 5
      A7 = 24 - 7
      A9 = 24 - 9
      A10 = 24 - 10
      NTRY = 100    }
      NTRY = 10     }
      NTRY = 1000   }
      NTRY = 2      }  Number of Trials Ⓟ
      NTRY = 5      }
      NTRY = 20     }
      NTRY = 10     }
      NRI = 23  }
      NRJ = 23  }  matrix Size
C
C                                                            Set Parameter
      ICASE = 0
40    CONTINUE                                               Set the Transition
      ICASE = ICASE + 1                                          Probability
      PGO = PGOAR(ICASE)  Ⓖ
      PST = PSTAR(ICASE)
      IF(PGO .EQ. 0.) GO TO 5000
      WRITE(MQ,42) ICASE,PGO ,NTRY ,NBRN
42     FORMAT(1H1, ////,20X,* *******RESULTS FOR CASE NO. *,I4,* *******
     1******,////,20X,*   THYE VALUE OF TRANSITION PROBABILITY IS *,
     2  ,F8.4,  *  NUMBER OF TRIALS =*,I5,* BURNING DURATION IS*,I5,//)
      WRITE(MQ,31)DELX,DELT
31    FORMAT(1H0, *   DISTANCE INCREAMENT =*,F12.4,*  TIME INCREMENT =*,
     1F12.4)
      WRITE(MQ,32) WIND,PHI,PBI
32    FORMAT(1H0,*  WIND SPEED = *,F12.4,*  FIREBRAND ANGLE = *,F12.4,
     1*PROB OF BLDG IGNITIN BY FIREBRANDS =*,F8.4)
```

```
      WRITE(MO,33) DSTIG
33    FORMAT(1H0  , *   DISTANCE FOR INITIAL IGNITION =*,F12.4)
      AL = (1.-(1. - PGO)**(1./XNBRN))/PGO
      PGO = AL*PGO
      DO 21 I = 1,NRI
      DO 21 J = 1,NRJ
      ISUM(I,J) = 0
21    CONTINUE
      ISUMA = 0
      ISUMB = 0
      ISUMC = 0
      ISUMD = 0
      ISUME = 0
      ISUMF = 0
      IT = 0
      TIME = 0.
C
C
100   CONTINUE                                                  Start New Trial
      IT = IT + 1
C
      IF(IPVAR .NE. 1) GO TO 303
      ITEM = IT/NITPV           }
      ITM = IT - ITEM*NITPV -1  } (5)
      IF(ITM .NE. 0) GO TO 303
      DO 302 I = 1,NRI
      DO 302 J = 1,NRJ
      RPAR(I,J) = 1. + RPSG*(RANF(0) - 0.5)
      SPAR(I,J) = 1. + SPSG*(RANF(0) - 0.5)  (w)
      FPAR(I,J) = 1. + FPSG*(RANF(0) - 0.5)
      SPFAR(I,J) = 1. + SPFSG*(RANF(0) - 0.5)
C
      SPFAR(I, 9) = 0.                    }
      SPFAR(I,10) = 0.                    }
      SPFAR(I,11) = 0.                    }
      SPFAR(I,12) = 0.                    }  Enter Firebrand
      SPAR(I, 9) = 0.                     }
      SPAR(I,10) = 0.                     }
      SPAR(I,11) = 0.                     }
      SPAR(I,12) = 0.                     }
302   CONTINUE
      WRITE(MO,461)
461   FORMAT(1H1,///20X,* ((((((((((RADIATION OUTPUT FACTOR*
     1,* FOR NEXT RUNS))))))))))))))*,//)
      WRITE(MO,209) (J,J = 1,NRJ)
      DO 465 I = 1,NRI
      WRITE(MO,305) I,(  RPAR(I,J),J=1,NRJ)
305   FORMAT( 1H0, * I =*,I5,5X, 2?F5.2)
465   CONTINUE
      WRITE(MO,462)
462   FORMAT(1H1,///20X,* ((((((((((RADIATION SUSEPTIBILITY FACTOR*
     1,* FOR NEXT RUNS))))))))))))))*,//)
      WRITE(MO,209) (J,J = 1,NRJ)
      DO 466 I = 1,NRI
      WRITE(MO,305) I,(  SPAR(I,J),J=1,NRJ)
466   CONTINUE
      WRITE(MO,463)
463   FORMAT(1H1,///20X,* ((((((((((FIREBRAND OUTPUT FACTOR*
     1,* FOR NEXT RUNS))))))))))))))*,//)
      WRITE(MO,209) (J,J = 1,NRJ)
      DO 467 I = 1,NRI
```

4

```
      WRITE(MQ,305) I,(  FPAR(I,J),J=1,NRJ)
467   CONTINUE
      WRITE(MQ,464)
464   FORMAT(1H1,///20X,* ((((((((((((FIREBRAND SUSTIBILITY FACTOR*
     1,* FOR NEXT RUNS))))))))))))))*,//)
      WRITE(MQ,209) (J,J = 1,NRJ)
      DO 468 I = 1,NRI
      WRITE(MQ,305) I,( SPFAR(I,J),J=1,NRJ)
468   CONTINUE
303   CONTINUE
C
C                                                          Initial Ignitions
      DO 101 I = 1,NRI
      DO 101 J = 1,NRJ
      ISTATE (I,J) = 0
      TSTATE(I,J) = 0.
      ING(I,J( = 0
      JNG(I,J) = 0
101   CONTINUE
      DO 103 I = 1,NRI
      DO 103 J = 1,NRJ
      TEMP = J
      TEM = PST*(1. - TEMP*DELX/DSTIG)  (JJ)
      IF(RANF(0) .GT. TEM) GO TO 103
      ISTATE(I,J) = 1
      ING(I,J) = I
      JNG(I,J) = J
      TDELAY = 0.063333 - ALOG(RANF(0))*TDMN (H)
      TSTATE(I,J) = TDELAY
      IGNIT = 1
      GO TO 400
431   CONTINUE
103   CONTINUE
      ICNT = 0
      TIME = -0.5*DELT
C                                                          Initial Ignition
C
C                                                          Start New Time Period
105   CONTINUE
      ICNT = ICNT + 1
      TIME = TIME + DELT  (B)
      DO 111 I = 1,NRI
      DO 112 J = 1,NRJ
      IF(TSTATE(I,J) .LE. 0.) GO TO 150
      IF(TIME .LT. TSTATE(I,J))GO TO 150
C     CHECK IGNITIONS
      IGNIT = 2
      GO TO 400
432   CONTINUE
      TSTATE(I,J)  = -TSTATE(I,J)
      ISTATE(I,J) = -1
150   CONTINUE
C
123   CONTINUE
124   CONTINUE
112   CONTINUE
111   CONTINUE
C
C
      IF(IPPTH .NE.1) GO TO 206  (B)
      IF(IT   .GT. NPPTH)  GO TO 206   (D)
      IF(ICNT .EQ.1) GO TO 447
      ITM = ICNT /NPPS              }(B)
      ITEMP = ICNT - NPPS*ITM     }
      IF(ITEMP .NE. 0) GO TO 206
447   CONTINUE
      WRITE(MQ,207) ICNT,TIME
207   FORMAT(1H1,///,*/////RESULTS AFTER STEP NO*,I5,*/////**   TIME = *
     1 ,F10.4,//)
      WRITE(MQ,209) (J,J = 1,NRJ)
      DO 205 I = 1,NRI
      WRITE(MQ,211) I,(ISTATE(I,J),J = 1,NRJ)
205   CONTINUE
      IF(IPTIG .NE. 1) GO TO 206 (C)
      WRITE(MQ,441)
441   FORMAT(1H1,//,10X,*   TIME TO IGNITE MATRIX*   ,///)
      WRITE(MQ,209) (J,J = 1,NRJ)
      DO 442 I = 1,NRI
      WRITE(MQ,443) I,(TSTATE(I,J),J = 1,NRJ)
443   FORMAT(1H0, * I =*,I5,5X,23F5.0)
442   CONTINUE
206   CONTINUE
C
C
      IDONE = 1                                                   Updat matrix
      DO 201 I = 1,NRI
      DO 201 J = 1,NRJ
      IF(TSTATE(I,J) .GT. 0.) IDONE = 0
      IF(ISTATE(I,J) .GT. 0 .AND. ISTATE(I,J) .LT.NBRN ) IDONE = 0
201   CONTINUE
      IF(IDONE .EQ.1) GO TO 223
      DO 202 I = 1,NRI
      DO 202 J = 1,NRJ
      IF(ISTATE(I,J) .LE.0) GO TO 203
      ISTATE(I,J) = ISTATE(I,J) + 1
203   CONTINUE
      IF(ISTATE(I,J) .EQ.-1) ISTATE(I,J) = 1
202   CONTINUE
      GO TO 105                                                  Updat Metrix
223   CONTINUE                                                 Start new Time Period
C
C
C
      DO 221 I = 1,NRI
      DO 221 J = 1,NRJ
      IF(ISTATE(I,J).EQ.0) GO TO 221
      ISUM(I,J) = ISUM(I,J) + 1
      IF(I .LT.  1.OR. I .GT.   A1) GO TO 251
      IF(J .LT.  1.OR. J .GT.   A1) GO TO 251
      ISUMA = ISUMA + 1
      IF(I .LT.  3.OR. I .GT.   A3) GO TO 251
      IF(J .LT.  3.OR. J .GT.   A3) GO TO 251
      ISUMB = ISUMB + 1
      IF(I .LT.  5.OR. I .GT.   A5) GO TO 251
      IF(J .LT.  5.OR. J .GT.   A5) GO TO 251
      ISUMC = ISUMC + 1
      IF(I .LT.  7.OR. I .GT.   A7) GO TO 251
      IF(J .LT.  7.OR. J .GT.   A7) GO TO 251
      ISUMD = ISUMD + 1
      IF(I .LT.  9.OR. I .GT.   A9) GO TO 251
      IF(J .LT.  9.OR. J .GT.   A9) GO TO 251
      ISUME = ISUME + 1
      IF(I .LT. 10.OR. I .GT.  A10) GO TO 251
      IF(J .LT. 10.OR. J .GT.  A10) GO TO 251
      ISUMF = ISUMF + 1
251   CONTINUE
221   CONTINUE
C
      IF(IPALL .NE.1) GO TO  215  Ⓐ
      IF(IT .GT. NTRYP) GO TO  215 Ⓕ
      WRITE(MQ,208) IT     ,ICNT ,PGO
208   FORMAT(1H1, ///////,20X,*==========RESULTS FOR TRIAL NO.*,I5,*=====
     1=========,/,*  RESULTS AFTER*,I6,* STEPS*,  *  TRANSITION PROB. =
     2*, F10.4)
      WRITE(MQ,209) (J,J = 1,NRJ)
209   FORMAT(1H0, 4X, * J = *    ,5X, 23I5,//)
      DO 210 I = 1,NRI
      WRITE(MQ,211) I,(ISTATE(I,J),J = 1,NRJ)
211   FORMAT( 1H0, * I =*,I5,5X, 23I5)
210   CONTINUE
      WRITE(MQ,455)
455   FORMAT(1H1,//,10X,* ............IGNITING MATRIX--NEGATIVE IS FIREB
     1RAND IGNITION .......*,//)
      WRITE(MQ,209) (J,J = 1,NRJ)
      DO 456 I = 1,NRI
      WRITE(MQ,457)I,(ING(I,J),JNG(I,J),J = 1,NRJ)
457   FORMAT(1H0, * I =*,I5,5X,23(I3,I2))
456   CONTINUE
215   CONTINUE
      IF(IT .LT. NTRY ) GO TO 100          Ⓟ
C
C
      WRITE(MQ,231) IT   ,PGO ,PST
231   FORMAT( 1H1,///, 20X,*-----------SUMMARRY RESULTS FROM*,I5,*TRIALS
     1---------------*,/,*      TRANS. PROB. =*,F10.4,*   IGNITION PROB. =
     2 *,F10.4,//)
      WRITE(MQ,209) (J,J = 1,NRJ)
      DO 232 I = 1,NRI
      WRITE(MQ,211) I,(  ISUM(I,J),J = 1,NRJ)
232   CONTINUE
      TMP = ISUMA
      TMA = TMP/529
      TMP = ISUMB
      TMB = TMP/361
      TMP = ISUMC
      TMC = TMP/225
      TMP = ISUMD
      TMD = TMP/121
      TMP = ISUME
      TME = TMP/49
      TMP = ISUMF
      TMF = TMP/25
252   FORMAT(1H0 ,* SUMS A-B-C-D-E-F = *,6I10)
      WRITE(MQ,252)ISUMA,ISUMB,ISUMC,ISUMD,ISUME,ISUMF
      WRITE(MQ,253) TMA,TMB,TMC,TMD,TME,TMF
253   FORMAT(1H0, *  FRACTIONS A-B-C-D-E-F = *,6F10.4)
      GO TO 40
C                                                          Set New Transition Probabilities
C
5000  CONTINUE
      STOP 6400
C
C
```

Radiation Ignition 7

```
C     POINT JUST IGNITED,SEE IF ITI WILL IGNITE OTHERS
400   CONTINUE
C     FIRST RADIATION
      DO 401 K = 1,4
      IF(K .NE. 1) GO TO 402
      IM = I-1
      JM = J
      GO TO 406
402   CONTINUE
      IF(K .NE. 2) GO TO 403
      IM = I + 1
      JM = J
      GO TO 406
403   CONTINUE
      IF(K .NE. 3) GO TO 404
      IM = I
      JM = J + 1
      GO TO -06
404   CONTINUE
      IF(K .NE. 4) GO TO 405
      IM = I
      JM = J - 1
      GO TO 406
405   CONTINUE
406   CONTINUE
      IF(IM .LE. 0 .OR .IM .GT. NRI) GO TO 401
      IF(JM .LE. 0 .OR .JM .GT. NRJ) GO TO 401
      IF(TSTATE(IM,JM) .LT.0.) GO TO 401
      TEMP = PGO*RPAR(I,J)*SPAR(IM,JM)   G
      IF(RANF(0) .GT. TEMP) GO TO 401
      TDELAY = 0.063333 - ALOG(RANF(0))*TDMN H  - Ignit Point and Find
      TST = TIME + TDELAY                          Time it can transmit burning
      IF(TSTATE(IM,JM) .GT. 0.)GO TO 408
      TSTATE(IM,JM) = TST
      ING(IM,JM) = I
      JNG(IM,JM) = J
      GO TO 409
408   CONTINUE
      IF(TST .GT. TSTATE(IM,JM))GO TO 409
      TSTATE(IM,JM) = TST
      ING(IM,JM) = I
      JNG(IM,JM) = J
409   CONTINUE
401   CONTINUE
C     NEXT FIREBRANDS
421   CONTINUE
      IF(J .EQ. NRJ) GO TO 420
C     FIND WHERE A POTENTIAL IGNITION MIGHT OCCUR
      ITM = 10.*RANF(0) + 1
      IF( ITM .EQ. 10) GO TO 422
      DIST = DISTAR(ITM) + DELAR(ITM)*(RANF(0) - 0.5)
      GO TO 423
422   CONTINUE
      DIST = 425. -84.*ALOG(RANF(0) )
423   CONTINUE
      DIST = DIST+0.25*WIND L I
      DCR = DIST*TNPH
      DCR = DCR*2.*(RANF(0) - 0.5)
      ITM = DIST/DELX   J
      JB = J + ITM
      IF(JB.LT. 1 .OR. JB.GT. NRJ) GO TO 425
```

Radiation Ignition

Firebrand Ignition

```
      ITM = DCR/DELX (3)
      IB = I + ITM
      IF(IB .LT. 1 .OR. IB .GT. NRI) GO TO 425
      IF(TSTATE(I,J) .LT.0.) GO TO 421
      TEMP = PBI*FPAR(I,J)*SPFAR(IB,JB)   (4)
      IF(RANF(0) .GT.  TEMP) GO TO 420
      TDELAY = 0.063333 - ALOG(RANF(0))*TDMN (4)
      TST = TIME + TDELAY
      IF(TSTATE(IB,JB) .GT. 0.)GO TO 428
      TSTATE(IB,JB) = TST
      ING(IB,JB) = -I
      JNG(IB,JB) = J
      GO TO 429
428   CONTINUE
      IF(TST .GT. TSTATE(IB,JB) ) GO TO 425
      TSTATE(IB,JB) = TST
      ING(IB,JB) = -I
      JNG(IB,JB) = J
429   CONTINUE
425   CONTINUE
      GO TO 421
420   CONTINUE
      GO TO(431,432) IGNIT
C
C
      END
```

COMPUTER PROGRAM DESCRIPTION

NAME: MESHFL

SYNOPSIS: Standard File Format Processing

TYPE: Production

USE: General purpose program for development and maintenance of Civil Defense data files.

BACKGROUND: Developed to complement and assist Standard File Format development.

DESCRIPTION: The program has a group of subprograms to do introduction of data files into standard format, merging of a set of data files, file editing, etc.

INPUT: Up to 10 data files

OUTPUT: Completed data file

STORAGE: IDA Card Deck #207

DOCUMENTATION: Attached description

LANGUAGE/SYSTEM: FORTRAN/6400 Scope

COMMENTS: Programs operational but not completely up to specification. Conversion to other machines needs changing number of BCD characters per machine word from 10 to the appropriate number.

A. INTRODUCTION

The program MESHFL was developed as a general utility program to manipulate standard format files. Chapter IV contains a description of these files; it is assumed here that the reader is generally familiar with them. It is also assumed in this description that records are identified by FIPS code; however, any other alphanumeric identifiers could be used--RSAC code, etc. The program has extensive internal documentation including definitions of all significant variables. This section then will only contain a general description of the program and its signfiicant subroutines, and a general description of the mode of using it. Due to the internal documentation in the program, no program annotations will be made. The program has several different functions, each performed by an individual subroutine. The first input card (in /, 6I10 format) reads the following parameters; if any are 1, that subroutine is called. At present only 1 function can be performed with each execution of the program.

IMRG Call MERGE to merge several input files into a single output file.

INTRO Call MEET to convert an input file into standard format.

IEXTRCT Calls EXTRCT to extract records from a file (not yet implemented).

IPNCH Calls PUNCH to put records in card image format.

IEDIT Calls EDIT to do editing on an input file.

IORDR Calls ORDER to add next record type and record count to put a file in complete standard format (not yet implemented).

The other data read in the main program are the variables IBUGA, ..., IBUGH (in /, 8I1 format). These switches turn on various types of debugging printout. They should normally all be turned off by having values 0.

The program requires standard input and output media to be defined. In addition input files MAT, MBT, MCT, etc., may be required for certain runs. Output is on the file TOUT. The program does not care on what physical medium the files are located, it being assumed this is defined by the operating system control. The input records must be BCD characters and no longer than 136 characters long, i.e., readable by standard FORTRAN statements.

This program has been used in the development of the standard files described in Chapter 4, in fact program development and file generation were carried out simultaneously. It was purposely written in standard FORTRAN to simplify transfer between machines. In the absence of a powerful system editing capability, this program has proved to be a very powerful tool in file development. Even with a computer with a powerful general purpose file editing capability, it is felt that the special purpose features of this program will make it more desirable to use in many situations in treating standard format files. There is clearly room for additional file handling capabilities to be included which would further enhance its capabilities. Two such items, clearly, are the use of this program in an interactive mode, and the addition of a random access file capability to the sequential file capability here.

B. SUBROUTINE MEET

This subroutine reads records in any format and places them into the standard format. The listing starts on page 25 and continues to page 33. Since the coding depends upon the format of the input records, no general purpose coding seemed worthwhile and special patches were written for each type of input used. After a file was safely converted, then presumably these special purpose codes should be discarded.

The subroutine first reads the variable INTRN (in/, I10 format) which selects the type of input tape. Control transfers then directly to the section of interest. The three samples here each read into an input buffer, and write from an output buffer. This is done so the next record type can be inserted based on input buffer information before writing from the output buffer. An alternative would be to use the subroutine order to perform this task. These three sections, beginning with statement numbers 211, 251, and 271 are all direct simple special purpose pieces of coding with no special interest to any general reader. They were, in fact, used for generating some of the standard format files described in Chapter 4.

C. SUBROUTINE MERGE

This subroutine merges several input files into a single output file. It is listed on pages 7 to 23. The subroutine assumes all input files are in FIPS code order. In can accommodate various numbers of records (up to 10 types) for each FIPS code in up to 10 input files, and missing records. Several options allow different ways of handling missing FIPS codes on input files. Once read, the records within a FIPS code may be output in any order.

The merge process is accomplished by using two basic buffers--a "next" buffer which contains data just input, and a "current" buffer which contains data to be output. When an input file is read, data is read into a "read" buffer and then transferred to the "next" buffer until a new FIPS code is encountered. Thus all the records from a single file in the "next" buffer are from a single FIPS code, although "next" buffers from different files may have different FIPS codes. Control parameters select the next FIPS code to be written on the output file, and all records from the "next" buffer are moved to the "current" buffer. They are written from the "current" buffer onto the output files in an order specified by control parameters. Before writing the appropriate next record, indicators, etc., are added. If more than one record of a particular type is present on an input file for a special FIPS code, the additional records are placed in a special buffer. They are saved then until the first of the group of multiple records is written from the "current" buffer, at which time the additional records are taken from the special buffer and output.

The first action of the subroutine is to read the control parameters ILEAD, ILDR, IADVL, IADVX, ISC, IVRY, IWTLSP, ITWLST, IWTMS, IFBGEN (in /, 5I10, /, 5I10, /, I3 format). The meaning and use of these parameters is given on page 2 of the program listing.

Next the number of input files, NMRGT, is read. Following this for each input file the number of records, NMRGA(I), is read. Then for this file and each record type, J, the four BCD characters identifying the record type NTYPA(I,J) are read, followed by the number of data characters in the record NCHRA(I,J) and up to 40 characters of description of the record which will identify it in output file references. The number of characters is required since on the output file only the number of data characters required for a given record type is written, rather than the full 110 possible. This can result in an efficient shortening of some output files. The output order is defined by specifying for the L^{th} record in the output file for any FIPS code the input file number in the array NORD(I), and the record type number for that input file NORD(J). Following this, control arrays for describing missing, INTMSN(I), or not used, IWTBDN(I), records are read. The format for these read statements is clear from the coding on figures 8 and 9.

Following the input parameters, a description of merging order is written. This is given by the code on pages 9 to 12.

The starting procedure is controlled on the bottom of page 12, which reads into the next buffer and finds a new FIPS code. Then the following cyclic procedures is followed:

Statement 120 transfer data from "next" buffer to
"current" buffer

Statement 130 read new data into next buffer

Statement 100 find "current" buffer FIPS code

Statement 140 write data from "current" buffer to output

The process is repeated until end of files on all necessary input files are sensed. For more detailed analysis of a section, the variable descriptions at the beginning of the program should first be read from which the fine structure in the coding should be clear.

D. SUBROUTINE EDIT

This subroutine edits records of a file in standard format. It is listed on pages 35 to 45. The following list describes the editing options available.

1--Delete a record of given FIPS code and type

2--Insert a record at specified FIPS code in proper order of types

3--Change a record to data specified

4--Insert a record after given FIPS code and type

5--Insert a record before given FIPS code and type

6--Move a record backwards in file

7-20--Change a record from one character to another

2X--Change record by code

An option allows the same change to be made for each record in a state. This subroutine has been used for some editing changes, but has not been completely checked for all specified capabilities.

The input of data for the edit subroutine will be described in order of execution of the read statements. The description starts with those on pages 36 of the program printout.

The first read is for number of record types, NTYPT, ISC as in defined in MERGE, and IPEDT=1 sets printing on standard output of the editing corrections. In the next read statement, the record label, the number of data characters in the record, and record description are read for each record in the order types specieied for a FIPS code. These data items are the same as in MERGE. This is necessary if items specified in location by read type (change type 2) are to be inserted in the proper place in the file.

The next set of reads are for common change items. ICOMNU of these are read. These are specified by a change reference number, ICOMNU. The first data character to change is ICOMFS(I). All characters are changed to ICOMLS(I). The data inserted is ICOMDA(I,L),L=1,7. Any time a change of type ICOMNU is specified in the charge list to be read later, all characters in the specified range have changed to the characters input here.

The next set of data items to read are state editing items. Those are NST of these items. The data read fills the arrays ISTLST(I) and ISTTP(I). For state with number ISTLST(I) all records are given the change LSTTP(I). Thus, for example, if the type change for state number I is 1, then all records for the state of Alabama would be delected from a file.

The next set of input read is similar to the state items but are for places. The number of changes is NPL, the place code to change is IPLLST(I), and the type change is IPLTP(I). If IPLFLG(I) is 1, changes are made for all place records of the type specified, if 0 only for the first.

The next set of data read is the individual editing items. There are a total of NSPNU of these reads, a maximum of 300 is allowed on any single run. For each record change, the card images are read. These contain all information for the record. Place values for the following arrays, which have the listed meaning in

JSPLST(I)--type of action to be taken,

JSPNFA(I)--state county FIPS code for editing action,

JSPNFB(I)--place FIPS code for editing action,

JSPNRT(I)--record type for correction.

The amount of valid data in those items depends upon the action to be taken; for example, if a record is to be inserted as an editorial action, then the entire record must be described; but if a record is to be deleted, no such description is needed.

After the data is read, then individual records from the file to be edited are read. The records progress through three buffers, the M buffer, which contains a record just read, the N buffer which contains a completed record except for the next record type, and the K buffer which contains a record ready to write from the output file. As a general rule, after a record is read into the M buffer, a search of editing instructions is made to see if editing action is needed associated with the record in the M buffer, if so, this action is specified. If, for example, a record is to be inserted, it is placed into the N buffer. Before this can be done, however, the N buffer must be cleared. This is done by writing from the K buffer and then advancing the record in the N buffer to the K buffer. The general procedure is one of pushing records through. An action is not performed until it is needed to clear a buffer to allow a record from an earlier buffer to advance. The action is always initiated by attempting to clear the M buffer so another record can enter the system from the input file.

```
      PROGRAM MESHFL(INPUT,OUTPUT,MAT,MBT,MCT,MDT,      TOUT,PUNCH)
C
C
C
C     A PROGRAM TO MESH SEVERAL FILE INTO A SINGLE FILE
C     ASSUMES RECORD TYPE AND FIPS CODE ON EACH RECORD
C     ADDS NEXT RECORD TYPE TO EACH RECORD
C     FEB., 1976, LEO SCHMIDT, I.D.A.  558-1346
C
C
C
C
C
C
C
C........BELOW ARE DEFINITIONS OF VARIABLES.........
C
C
C
C
C...INPUT CONTROL AND FILE DEFINITION VARIABLES
C
C
C...CONTROL OF TYPE
C
C     IMRG-- IF 1 MERGE SEVERAL FILES
C     INTRO-- IF 1 CONVERT A TAPE FROM SOME FORMAT TO STD. FORMAT
C     IEXTRT-- IF 1 EXTRACT RECORDS FROM A FILE
C     IEDIT-- IF 1 DO EDITING ON AN INPUT FILE
C     IORDR-- PUT FILE IN COMPLETE STANDARD FORM BY ADDING NEXT
C     RECORD TYPE AND RECORD COUNT, NO OTHER CHANGES
C     IPNCH-- PUNCH ON CARDS RECORDS INDICATED BY CODE TYPE AND FIPS
C        CODE.
C     IBUGA-- VARIABLE TO CONTROL DEBUG PRINTOUT, CONTROL OF PRINTOUT OF
C        WRITTEN OUTPUT TAPE DURING EXECUTION
C
C...CONTROL OF BASIC FORMAT SETTING
C
C     INTRN-- NUMBER OF TYPE TO CONVERT IN GO TO STATEMENT
C        1 ANCET CITY DATA BASE
C        2 CCDB SELECTIONS
C        3 BLAST AND FALLOUT RISK
C
C...CONTROL OF EXTRACT
C
C     IGEOG-- EXTRACT RECORDS BASED ON LOCATION
C     IPOLIT-- EXTRACT RECORD ON A POLITICAL BASIS
C     IDELT-- DELETE NAMED RECORD TYPE
C     ISAVE-- SAVE NAMED RECORD TYPE
C
C
C...CONTROL OF EDITING ROUTINE
C
C        JFBSP(300)-- AND REST OF SPECIAL BUFFER VARIABLES ARE SET OF
C        CORRECTIONS INPUT FROM STANDARD INPUT TO CONTROL EDITING OF
C        FILE INPUT ON MAT
C     JSPLST(300)-- ARRAY FOR CODE ON TYPE OF EDITING
C        1 DELETE A RECORD WITH GIVEN FIPS CODE AND TYPE
C        2 INSERT A RECORD SO CODE AND TYPE ARE IN ORDER
C        3 CHANGE A RECORD TO THAT GIVE IN CORRECTION
C        4 INSERT RECORD AFTER GIVEN FIPS CODE AND TYPE
```

```
C         5 INSERT A RECORD BEFORE GIVEN CODE AND TYPE
C         FOR 4 AND 5 TYPES JSPNFA,JSPNFB, AND JSPNRT ARE CODE AND TYPE
C         OF INSERTED RECORD
C         6 MOVE A RECORD FROM GIVEN CODE AND TYPE LOCATION
C            TO FIPS CODE AND TYPE GIVEN BY JSPNFA, JSPNFB, JSPNRT INPUT
C            AFTER DATA
C         1X DO STANDARD CHANGE OF TYPE 1X - 10, CHANGES DEFINE BY CODE
C            INSERTED BY USER
C      NST-- NUMBER OF STATE WITH GROUP TREATMENT IN EDITING
C      ISTLST(50)-- LIST OF STATES WITH SUCH TREATMENT
C      ISTTP(20)-- TYPE OF TREATMENT FOR THE STATE
C      ISC-- AS WITH MERGE, FIPS CONTROL
C      IPEDT-- IF 1 LIST EDITING CORRECTIONS TO MAKE
C      ICOMDA(20,10)-- DATA FOR COMMON CHANGES, TO BE INSERTED
C      ICOMFS(20)-- FIRST CHARACTER OF COMMON CHANGE
C      ICOMLS(20)-- LAST CHARACTER OF COMMON CHANGES
C      ICOMTP(40)-- INDEX TO LOCATION WITH TYPE OF CHANGE
C      ICOMNU-- NUMBER OF COMMON CHANGE TYPES
C
C...CONTROL OF PUNCH
C
C      ISC-- CONTROL OF FIPS CODE USE, AS IN MERGE
C      IPRT-- IF ONE OUTPUT THE RECORDS PUNCHED
C
C...CONTROL OF MERGE
C
C      IWTMS-- 1 WRITE ON OUTPUT FILE MISSING FIPS CODE AND RECORD TYPE,
C         2 WRITE ON TAPE, 0 DONT
C      IWTMSN(20)-- IF RECORD NUMBER IS ONE OF THESE DO NOT RECORD IF
C         MISSING EVEN IF IWTMS IS 1 OR 2
C      NWTMS-- NUMBER OF RECORD TYPES NOT TO RECORD
C      IWTBDP-- IF ONE WRITE CODE TYPE FILE FOR RECORD WITH TYPE NOT IN
C         LIST OF ACCEPTABLE TYPES , IF 2 WRITE DATA, IF 0 DONT DO
C         PUT ON FILE OUTPUT
C      IWTBDT-- AS AS ABOVE BUT ON FILE POUT
C      IWTBDN(20)-- IF BAD RECORD ONE OF THESE TYPES DONT WRITE
C      NWTBD-- NUMBER OF THESE RECORDS NOT TO WRITE
C      IWTLSP-- IF 1 WRITE ON OUTPUT RECORDS NOT USED, JUST FIPS CODE,
C         TYPE, AND RECORD FOLLOWING, IF 2 WRITE DATA ALSO
C      IWTLST-- SAME AS IWTLSP BUT WRITE ON FILE POUT
C      IFBGEN-- 1 GENERATE SPACES FOR NEW CNTY, 2 NEW PAGE FOR STATE ALSO
C         3 NO SPACE, 0 LEAVE ALONE.
C      ILEAD-- IF 0 USE LOWEST CODE ON NEXT FILES AS NEXT FIPS CODE
C         GROUP TO PRINT, IF NOT ZERO USE THE CODE WITH THE  ILEAD TH
C         FILE. NOTICE ILEAD 0 ASSUMES THE FILE CODES ARE IN INCREASING
C         NUMERICAL ORDER. ALSO ALL RECOORDS ARE USED.
C      IADVL-- IF 1 ADVANCE ILEAD FILE IF THE ILDR RECORD IS NOT PRESENT
C         IF 0 DO NOT
C      ILDR-- THE RECORD NUMBER FOR FILE ILEAD ADVANCING CONTROL
C      IADVX-- IF 1 ASSUME FILES ARE IN NUMERICAL ORDER,IF ILEAD FILE
C         HAS LARGER VALUE ADVANCE OTHER FILES TILL THEY MATCH AND
C         PASSED OVER RECORDS, IF 2 ADVANCE OTHER FILES TILL MATCH, NO
C         NUMERICAL ORDERING ASSUMED, NOTE THIS MAY CAUSE FILE TO ADVANCE
C         TILL EOF . IF 3 DO NOT ADVANCE OTHER FILES UNLESS A MATCH IS
C         FOUND IN READ BUFFER. ANY OTHER VALUE ACTS LIKE 3
C      IVRY-- IF 1 A RECORD TYPE MAY BE REPEATED ON A FILE NEEDING
C         SPECIAL HANDLING BY A BUFFER
C      ISC-- IF 1 COMBINE ALL STATE AND COUNTY RECORDS TOGETHER IN A
C         GROUP, IF 0 PLACE REORDS TOO DEFINE A GROUP
C
C      NWRGT-- NUMBER OF EXTERNAL FILES TO MERGE
```

```
C     NMRGA(10)-- NUMBER OF TYPES IN FILE I
C     NTYPA(10,10)---- A4 TYPE CODE FOR J TH RECORD IN THE ITH FILE
C     NCHRA(10,10)-- I,J NO OF CHARACTERS IN J RECORD OF FILE I
C     NDESCR(10,10,4)-- DESCRIPTION OF I J TH FILE
C
C
C...DERIVED RECORD/FILE STRUCTURE VARIABLES
C
C     NORDI(100)-- FILE I RECORD J IS THE K TH RECORD TO BE WRITTEN ON
C        THE MERGED FILE I3
C     NORDJ(100)-- SAME FOR J
C     NTYPIJ(100)-- TYPE OF IJ TH RECORD
C     NCHRCU(100)-- NO OF CHARACTERES IN L TH FILE.
C     NLNKL(10,10)-- TYPE I RECORD J IS NLNKL IN BUFFER
C     NTYPT-- TOTAL NUMBER OF TYPES OF RECORDS TO MERGE
C
C
C...CURRENT BUFFER VARIABLES
C
C     JFBCU(100)-- FORMAT CONTROL BIT IN CURRENT BUFFER
C     JRECTP(100)--  RECORD TYPE OF CURRENT FIPS BUFFER
C     JNXTP(100)-- NEXT RECORD OF CURRENT FIPS BUFFER
C     JRCTCU(100)-- READ RECORD COUNT IN  CURRENT BUFFER
C     JCUFPA(100)--  FIPS CODE IN CURRENT BUFFER, PART A
C     JCUFPB(100)-- SAME, PART B
C     JCURDA(100,15)-- DATA IN CURRENT BUFFER
C     ICF(100)-- FLAG FOR CURRENT BUFFER, 0 INVALID DATA, 1 CURRENT FIPS
C        CODE DATA
C     ICUFPA -- VALUE OF CURRENT BUFFER, PART A, IE FIPS CODE
C        STATE AND COUNTY
C     ICUFPB-- VALUE OF CURRENT BUFFER, PART B,PLACE
C
C
C...NEXT BUFFER VARIABLES
C
C     JFBNX(100)-- FORMAT CONTROL BIT IN NEXT BUFFER
C     JNFCUR(100)-- RECORD TYPE OF NEXT FIPS BUFFER
C     JNFNXT(100)-- NEXT RECORD OF NEXT FIPS BUFFER
C     JRCTNX(100)-- READ RECORD COUNT IN NEXT BUFFER
C     JNBFPA(100)-- FIPS CODE IN NEXT BUFFER, PART A
C     JNBFPB(100)-- SAME, PART B
C     JNXTDA(100,15)-- DATA IN NEXT BUFFER
C     INF(100)-- FLAG FOR NEXT  FIPS BUFFER, 0 EMPTY NEEDS DATA,
C        1 VALID DATA FROM FILE READ, 2 END OF DATA ON THIS FILE,
C        DO NO MORE READING FROM IT
C     KCUFPA -- VALUE OF NEXT BUFFER, PART A, I.E. FIPS CODE
C        STATE AND COUNTY
C     KCUFPB-- VALUE OF NEXT BUFFER, PART B, PLACE
C     KFPNBA(10)-- FIPS CODE,PART A, OF ITH GROUP IN NEXT BUFFER
C     KFPNBB(10)-- PART B
C
C
C...SPECIAL BUFFER VARIABLES
C
C     JFBSP(300)-- FIRST BIT IN SPECIAL BUFFER
C     JSPCUR(300)-- CURRENT RECORD TYPE IN EXTRA BUFFER
C     JNFSP(300)-- NEXT RECORD IN EXTRA NUFFER
C     JRCTSP(300)-- RECORD COUNT IN SPECIAL BUFFER
C     JSPFPA(300)-- FIPS CODE PART A IN EXTRA BUFFER
C     JSPFPB(300)-- PART B
C     JSPDA(300,15)-- DATA IN EXTRA BUFFER
```

```
C      NSPNU-- NUMBER OF RECORDS IN SPECIAL BUFFER
C      ISPFST(300)-- FIRST RECOD NU OF L TH RECORD IN SPECIAL BUFFER,
C         IF 0 NONE PRESENT -- ASSOCIATED WITH CURRENT BUFFER VARIABLES
C      ISPLST(300)-- LAST RECORD IN SPEC BUFFER FOR L TH TYPE
C         FOR CURRENT BUFFER
C      JSPFST(300)-- AS I BUT NEXT BUFFER VARIABLES
C      JSPLST(300)-- AS I BUT NEXT BUFFER VARIABLES
C      ICUB-- IF 1 DRAW DOWN CURRENT BUFFER IF 2 NEXT BUFFER STORES IN
C         SPECIAL BUFFER DRAWDOWN
C      MXSPBU-- MAXIMUM NUMBER OF ENTRIES IN SPECIAL BUFFER
C
C
C...READ BUFFER VARIABLES
C
C      JFBHR(10)-- LIKE SPECIAL BUFFER NAMES; USED TO HOLD DATA FOR A
C         NEXT FIPS CODE READ ON FILE I. USED WHEN RECORDS ARE MISSING
C         TO TERMINATE NORMAL READS.
C      IHR(10)-- THERE IS DATA FOR ITH FILE READ TO BE INSERTED ON
C         NEXT READ
C      IRDF(10)-- FLAG TO TELL THAT JTH FILE HAS BEEN READ ON CURRENT
C         READ
C      KFPRDA(10)-- FIPS CODE,PART A, OF ITH GROUP IN READ BUFFER
C      KFPRDB(10)-- SAME PART B
C
C
C...MISCELLANEOUS VARIABLES
C
C      I-- USUALLY INDEX TO INPUT FILE NUMBER
C      J-- USUALLY INDEX TO INPUT RECORD NUMBER
C      L-- USUALLY INDEX TO RECORDS IN ORDER OF OUTPUT
C      JRCDCT-- COUNT OF RECORDS WRITTEN  ON OUTPUT FILE
C      MRD(10)-- EXTERNAL NAMES OF READ FILE
C      MNXDA(15)-- DATA IN READ BUFFER
C      MXWRD-- MAXIMUM NO OF  DATA WORDS IN ONE READ
C      NCHRW-- NUMBER OF CHARACTERS/WORD
C      IFLEF(10)-- END OF FILE SENSED ON ITH FILE TYPE
C      MFBNX,MNBFPA,MNBFPB,MNFCUR,MNFNXT,MRCTNX,MNXDA(15)-- DATA IN
C         TEMPORARY READ BUFFER
C      ISTRT-- FLAG TO INDICATE START OF FILE MERGE FOR INITIAL FILL OF
C         NEXT BUFFER
C      ISTOP-- FLAG TO INDICATE ALL DATA HAS BEEN READ
C      NTSTP-- VALUE TO WRITE IN LAST RECORD ON MERGED FILE CURRENT VALUE
C         IS  STOP
C      NTST-- FOR DEBUGGING NUMBER OF RECORDS TO PRODUCE BEFORE ENDING
C         RUN. FOR PRODUCTION RUNS SET TO A LARGE NUMBER
C      NWRD-- COMPUTED NUMBER OF DATA WORDS TO WRITE ON OUTPUT RECORD
C         COMPUTED FROM NCHRCU
C      MP-- STANDARD INPUT FILE, FOR DATA AND CONTROL
C      MQ-- STANDARD OUTPUT FILE
C      MS-- PUNCH FILE
C      MOP-- PRINTED OUTPUT- EXTRA
C      MOT-- OUTPUT TAPE FILE FOR MERGED RECORDS
C
C
C...VARIABLES USED IN EDITING FOUTINE
C
C      JSPFST(300)-- IF ONE THIS CORRECTION HAS BEEN USED, DONT SEARCH
C         IT ANY MOFE
C      ISPFST(300)-- IF 1 INSERT LTH CORRECTION AFTER NEXT WRITE,
C         IF 2 AFTER SECOND WRITE
C      NTYPIJ(100)-- AS IN MERGE IS LTH RECORD TYPE
```

```
C     NCHRCU(100)-- AS MERGE, NUMBER OF CHARCTERS OF DATA IN TYPE
C     ISPLST(300)-- PLACE OF RECORD IN LIST
C     MBUFFL-- IF 1 M BUFFER HAS VALID DATA, IF 0 NOT
C     NBUFFL-- VALID DATA FLAG FOR N BUFFER
C     KBUFFL-- VALID DATA FLAG FOR K BUFFER
C     NIST-- NUMBER OF INSERTS AFTER NEXT RECORD WRITE
C     NSPNU-- NUMBER OF INDIVIDUAL CORRECTIONS TO MAKE
C     JDOC-- IF 1 MAKE CHANGE CORRECTION(TYPE 3)
C     JDOG-- IF 1 MAKE COMMON CHANGE TYPE JDOG - 10
C     MFBNX-- ETC. M, N, OR K BUFFER VARIALBE, AS NEXT BUFFER BUT
C        CHANGE FIRST LETTER AND NOT ARRAYS
C
C
C
C
      COMMON / FILOR/ NMRGA(10), NTYPA(10,10) , NORDI(100), NORDJ(100),
     1 NTYPIJ(100), NLNKL(10,10), NDESCR(10,10,4)  , NCHRA(10,10),
     2   NCHRCU(100),NMRGT,NTYPT
C
      COMMON / FNBUF/ JFBCU(100),JRECTP(100), JNXTP(100), JRCTCU(100),
     1 JCUFPA(100), JCUFPB(100),JCURDA(100,15),ICF(100)
     2, JFBNX(100), JNFCUR(100),JNFNXT(100), JRCTNX(100), JNBFPA(100),
     3 JNBFPB(100),JNXTDA(100,15),INF(100)
C
      COMMON /SPBUF/ JFBSP(300), JSPCUR(300), JNFSP(300),JRCTSP(300),
     1 JSPFPA(300), JSPFPB(300), JSPDA(300,15),NSPNU,
     2 ISPFST(300), ISPLST(300), JSPFST(300), JSPLST(300)
C
      COMMON / CONS/ IBUGA,IBUGB,IBUGC,IBUGD,IBUGE,IBUGF,IBUGG,IBUGH
     1  , MSKAR(10), MSKA,MSKB,MSKC,MSKD, MSKE,MSKF, MP,MQ,MS,MOT,MOP,
     2 MRD(10),MTA,MTB,MTC,MTD,MTE,MTF,MTG,MTH,MTI,MTJ
     3 ,MXWRD,NTSTP,NCHRW ,MXSPBU
C
      COMMON /MISC/ KNXDA(15), MNXDA(15), NNXDA(15), ICNT,NTST,ISTOP
     1,JRCDCT
C
C
C     6400 UNIQUE
      DATA  MSKAR     / 77B,7700B,770000B,77000000B,7700000000B,
     1 770000000000B  ,77000000000000B      ,7700000000000000B,
     2 770000000000000000B,  77000000000000000000B/
      DATA  MSKA  /   77777777777777777777B  /
      DATA  MSKB /  33000000000000000000B/
      DATA  MSKC /  77000000000000000000B/
      DATA  MSKD /  55000000000000000000B/
      DATA  MSKE /  77770000000000000000B/
      DATA  MSKF /  77B/
C
C     TO CORRESPOND TO INPUT FILES ON PROGRAM CARD
      DATA  MRD  / 3LMAT,3LMBT,3LMCT,3LMDT,3LMET,3LMFT,3LMGT,
     1 3LMHT,3LMIT,3LMJT/
      DATA   MTA / 3LMAT/
      DATA   MTB / 3LMBT/
      DATA   MTC / 3LMCT/
      DATA   MTD / 3LMDT/
      DATA   MTE / 3LMET/
      DATA   MTF / 3LMFT/
      DATA   MTG / 3LMGT/
      DATA   MTH / 3LMHT/
      DATA   MTI / 3LMIT/
      DATA   MTJ / 3LMJT/
      DATA  MP  /5LINPUT /
      DATA  MQ  / 6LOUTPUT/
      DATA  MS / 5LPUNCH/
      DATA  MOT   /4LTOUT/
      DATA  MOP  / 4LPOUT/
C
C
C
C
C.......START EXECUTTABLE CODE............
C
C
C     INITIALIZE
      NTST = 5
      NTST = 10
      NTST = 100
      NTST = 200
      NTST = 9999999
      NTST = 50
      NTST = 20
      ICNT = 0
C
      NTSTP = 4HSTOP
      ISTOP = 0
      JRCDCT = 0
C
C
      WRITE(MQ,8)
8     FORMAT(1H1,///,30X,* DATA FROM RUN OF FILE MERGE PROGRAM*,//////)
C
C     AGAIN FOR 6400 NCHRW = 10
      NCHRW = 10
      MXWRD = 11
C
      NSPNU = 0
      MXSPBU = 300
      DO 4 L = 1,300
      ISPFST(L) = 0
      ISPLST(L) = 0
      JSPFST(L) = 0
      JSPLST(L) = 0
4     CONTINUE
C
C
C
C     READ CONTROL PARAMETERS
C
      READ(MP,11) IMRG,INTRO,IEXTRT,IPNCH,IEDIT,IORDR
11    FORMAT(/,6I10)
      READ(MP,9) IBUGA,IBUGB,IBUGC,IBUGD,IBUGE,IBUGF,IBUGG,IBUGH
9     FORMAT(/,8I1)
      IF(IMRG   .NE. 1) GO TO 21
      CALL MERGE
      GO TO 1500
21    CONTINUE
      IF(INTRO  .NE. 1) GO TO 22
      CALL MEET
      GO TO 1500
22    CONTINUE
      IF(IEXTRT .NE. 1) GO TO 23
      CALL EXTRCT
      GO TO 1500
23    CONTINUE
      IF(IPNCH  .NE. 1) GO TO 24
      CALL PUNCH
      GO TO 1500
24    CONTINUE
      IF(IEDIT  .NE. 1) GO TO 25
      CALL EDIT
      GO TO 1500
25    CONTINUE
      IF(IORDR  .NE. 1) GO TO 26
      CALL ORUER
      GO TO 1500
26    CONTINUE
C
C
1500  CONTINUE
      STOP 6400
      END
      SUBROUTINE MERGE
C
C
C........................................................................
C                            BEGIN FILE MERGE SECTION
C                                 *      .      +
C                                  *     .     +
C                                   *    .    +
C                                    *   .   +
C                                     *  .  +
C                                      * . +
C                                       *.+
C                                        =
C                                        =
C                                        =
C
C
C
C     THIS SECTION MERGES SEVERAL FILES.
C     DATA IS READ INTO THE NEXT BUFFER. DATA IS WRITTEN FROM THE
C     CURRENT BUFFER. USUALLY ALL DATA FROM A STATE/COUNTY IS
C     WRITTEN AT ONCE IN THE ORDER SPECIFIED BY INPUT. AFTER WRITING,
C     DATA IS ADVANCED FROM THE NEXT BUFFER INTO THE CURRENT BUFFER,
C     AND DATA FROM A NEW ST/CNTY IS READ INTO THE NEXT BUFFER.
C     IF MORE THAN 1 RECORD OF A TYPE IS ON A FILE, IVRY MUSRT BE
C     ONE. THEN THE EXTRA RECORDS ARE READ INTO A SPECIAL BUFFER AND
C     SAVED TO BE WRITTEN AFTER THE FIRST OF THE TYPE IS WRITTEN.
C     ALL RECORDS WITH A PARTICULAR CODE MUST BE ADJACENT ON A FILE.
C     THEY MAY BE IN ANY ORDER WITHIN THE GROUP OF RECORDS WITH THE SAME
C     CODE.
C
C
C
C
      DIMENSION           IFLEF(10),IWTMSN(20),IWTBDN(20)
      DIMENSION JFBHR(10),JHRCUR(10),JNFHR(10),JHRFPA(10),JHRFPB(10),
     1 JRCTHR(10), JHRDA(10,15),IHR(10),IRDF(10)
      DIMENSION KFPNBA(10),KFPNBB(10),KFPRDA(10),KFPRDB(10)
      COMMON / FILOR/ NMRGA(10), NTYPA(10,10) , NORDI(100), NORDJ(100),
     1 NTYPIJ(100), NLNKL(10,10), NDESCR(10,10,4)  , NCHRA(10,10),
     2   NCHRCU(100),NMRGT,NTYPT
      COMMON / FNBUF/ JFBCU(100),JRECTP(100), JNXTP(100), JRCTCU(100),
      1 JCUFPA(100), JCUFPB(100),JCURDA(100,I5),ICF(100)
      2, JFBNX(100), JNFCUR(100),JNFNXT(100), JRCTNX(100), JNBFPA(100),
      3 JNBFPB(100),JNXTDA(100,15),INF(100)
       COMMON /SPBUF/ JFBSP(300), JSPCUR(300), JNFSP(300),JRCTSP(300),
      1 JSPFPA(300), JSPFPB(300), JSPDA(300,I5),NSPNU,
      2 ISPFST(300), ISPLST(300), JSPFST(300), JSPLST(300)
       COMMON / CONS/ IBUGA,IBUGB,IBUGC,IBUGD,IBUGE,IBUGF,IBUGG,IBUGH
      1  , MSKAR(10), MSKA,MSKB,MSKC,MSKD, MSKE,MSKF, MP,MQ,MS,MOT,MOP,
      2 MRD(10),MTA,MTB,MTC,MTD,MTE,MTF,MTG,MTH,MTI,MTJ
      3 ,MXWRD,NTSTP,NCHRW ,MXSPBU
       COMMON /MISC/ KNXDA(15), MNXDA(15), NNXDA(15), ICNT,NTST,ISTOP
      1,JRCDCT
C
C
C
C      INITIALIZE
       KWTMS = 1
       ISTRT = 1
       ISTO = 2H
       DO 2 L = 1,100
       ICF(L) = 0
       INF(L) = 0
2      CONTINUE
       DO 3 I = 1,10
       IFLEF(I) = 0
       IHR(I) = 0
3      CONTINUE
C
C
C
C      READ MERGE CONTROL PARAMETERS
C
       READ(MP,11) ILEAD,ILDR,IADVL,IADVX,ISC
11     FORMAT(/,5I10)
       READ(MP,11) IVRY, IWTLSP,IWTLST,IWTMS,IFBGEN
       READ(MP,11) IWTBDP,IWTBDT
C
C      READ FILE DEINITION PARAMETERS
C
       READ(MP,12) NMRGT
12     FORMAT(/,I10)
       NTYPT = 0
       DO 13 I = 1,NMRGT
       READ(MP,14) NMRGA(I)
-14    FORMAT(I2)
       ITMP = NMRGA(I)
       NTYPT = NTYPT + ITMP
       READ(MP,15)(NTYPA(I,J), J = 1,ITMP)
15     FORMAT(10(1X,A4))
       READ(MP,16)(NCHRA(I,J),J = 1,ITMP)
16     FORMAT(10I5)
       DO 26 J = 1,ITMP
       READ(MP,27)(NDESCR(I,J,K),K = 1,4)
27     FORMAT(4A10)— 6400 unique
26     CONTINUE
13     CONTINUE
C
       DO 17 L = 1,NTYPT
       READ(MP,18) NORDI(L),NORDJ(L)
18     FORMAT(2I5)
       I = NORDI(L)
```

AD-A040 945 INSTITUTE FOR DEFENSE ANALYSES ARLINGTON VA PROGRAM --ETC F/G 15/4
DOCUMENTATION OF CURRENT IDA COMPUTER MATERIAL DEVELOPED FOR DC--ETC(U)
JAN 77 L A SCHMIDT DCPA01-76-C-0213

UNCLASSIFIED IDA/HQ-77-19225 NL

3 OF 3

AD
A040 945

END

DATE
FILMED
7 - 77

```
      J = NORDJ(L)
      NTYPIJ(L) = NTYPA(I,J)
      NLNKL(I,J) = L
      NCHRCU(L) = NCHRA(I,J)
17    CONTINUE
C
      NWTMS = 0
      IF(IWTMS .EQ.0) GO TO 21
23    CONTINUE
      READ(MP,22) ITMP
22    FORMAT(I3)
      IF(ITMP .LE.0) GO To 21
      NWTMS = NWTMS + 1
      IWTMSN(NWTMS) = ITMP
      GO TO 23
21    CONTINUE
C
      NWTBD = 0
      IF(IWTBDP .EQ.0 .AND. IWTBDT .EQ.0) GO TO 976
977   CONTINUE
      READ(MP,978)ITMP
978   FORMAT(A4)
      IF(ITMP .EQ. 4HZZZZ) GO TO 976
      NWTBD = NWTBD + 1
      IWTBDN(NWTBD) = ITMP
      GO TO 977
976   CONTINUE
C
C
C     PRINT RUN DESCRIPTION
C
      WRITE(MQ,181) NMRGT
181   FORMAT(1H0,20X,*IN THIS CALCULATION MERGE THE DATA FROM*,I3,
     1 * SEPERATE INPUT FILES*,/, 30X, *......................*,///)
C
      IF(IVRY .NE. 1) GO TO 185
      WRITE(MG,188)
188   FORMAT(1H0, * IF THIS RUN A RECORD TYPE MAY BE REPEATED, UP TO 300
     1 REPEATED RECORDS PER SET ARE ALLOWED*)
      GOTO 186
185   CONTINUE
      WRITE(MQ,189)
189   FORMAT(1H0, *NO REPEATED RECORD TYPES IN A SINGLE SET ARE ALLOWED
     1IN THIS RUN*)
186   CONTINUE
C
      IF(ILEAD .NE. 0) GO TO 182
      WRITE(MQ,184)
184   FORMAT(1H0, *USE LOWEST FIPS CODE IN INPUT FILES AS CODE FOR NEXT
     1SET OF RECORDS WRITTEN -- ALL INPUT RECORDS ARE USED*)
      GO TO 918
182   CONTINUE
      WRITE(MQ,187) ILEAD
187   FORMAT(1H0, * TAKE FIPS CODE FOR NEXT SET OF RECORDS AS THAT ASSOC
     1IATED WITH FILE NO.*, I3 )
183   CONTINUE
C
      IF(IADVL .NE. 0) GO TO 911
      WRITE(MG,913) ILEAD
913   FORMAT(1H0,* DO NOT SKIP ANY GROUPS OF RECORDS IN LEAD FILE*,I3)
      GO TO 912
```

```
911   CONTINUE
      WRITE(MQ,914)ILDR,ILEAD
914   FORMAT(1H0,* IF THE *,I3,* RECORD IS MISSING FROM A GROUP IN THE L
     1EAD FILE*,I3,* SKIP THIS GROUP OF RECORDS AND ADVANCE TO THE NEXT*
     2)
912   CONTINUE
C
      IF(IADVX .NE.1) GO TO 916
      WRITE(MQ,919)
919   FORMAT(1H0,*ASSUME FILES IN NUMERICAL ORDER-- IF LEAD FILE HAS LAR
     1GER FIPS CODE ADVANCE OTHERS AND DROP SKIPPED RECORDS*)
      GO TO 918
916   CONTINUE
      IF(IADVX .NE.2) GO TO 917
      WRITE(MQ,920)
920   FORMAT(1H0,* IF LEAD FILE CODE DIFFERENT FROM OTHERS,ADVANCE OTHER
     1S UNTIL MATCH IS FOUND*)
      GO TO 918
917   CONTINUE
      WRITE(MQ,921)
921   FORMAT(1H0,* DO NOT ADVANCE FILES TO MATCH CODES WITH LEAD FILE UN
     1LESS CODE IS IN READ BUFFER*)
918   CONTINUE
C
      IF(IWTMS.EQ. 1) GO TO 801
      IF(IWTMS.EQ. 2) GO TO 802
      WRITE(MQ,804)
804   FORMAT(1H0, * IF A RECORD TYPE IS MISSING FROM A FIPS CODE GROUPIN
     1G DO NOT RECORD THIS EVENT*)
      GO TO 803
801   CONTINUE
      WRITE(MQ,805)
805   FORMAT(1H0, * IF A RECORD TYPE IS MISSING FROM A FIPS CODE GROUPIN
     1G RECORD THIS ON OUTPUT FILE*)
      GO TO 902
802   CONTINUE
      WRITE(MQ,806)
806   FORMAT(1H0, * IF A RECORD TYPE IS MISSING FROM A FIPS CODE GROUPIN
     1G RECORD THIS ON TAPE FILE MOP*)
902   CONTINUE
      IF(NWTMS .EQ.0) GO TO 901
      WRITE(MQ,903) NWTMS
903   FORMAT(/,1H0, * DO NOT RECORD MISSING RECORDS FOR THE FOLLOWING*,
     1 I2,* TYPES OF RECORDS*,/ *  NO.*,2X, *POSN  TYPE   DESCRIPTION*,
     2 /)
      DO 904 K = 1,NWTMS
      L = IWTMSN(K)
      I = NORDI(L)
      J = NORDJ(L)
      WRITE(MQ,905) K,IWTMSN(K),NTYPIJ(L), (NDESCR(I,J,KK),KK= 1,4)
905   FORMAT(1H , I4,2X,I4,2X,A4,2X,4A10)
904   CONTINUE
901   CONTINUE
803   CONTINUE
C
      IF(IWTLSP .EQ. 1) GO TO 807
      IF(IWTLST .EQ. 1) GO TO 808
      IF(IWTLSP .EQ. 2) GO TO 817
      IF(IWTLST .EQ. 2) GO TO 818
      WRITE(MQ,813)
813   FORMAT(1H0,* IF AN INPUT RECORD IS NOT USED DO NOT RECORD THIS EVE
     1NT*)
      GO TO 809
807   CONTINUE
      WRITE(MQ,812)
812   FORMAT(1H0,* IF AN INPUT RECORD IS NOT USED RECORD CODE AND TYPE O
     1N OUTPUT FILE*)
      GO TO 809
808   CONTINUE
      WRITE(MQ,811)
811   FORMAT(1H0,* IF AN INPUT RECORD IS NOT USED RECORD CODE AND TYPE O
     1N TAPE FILE POUT*)
      GO TO 809
817   CONTINUE
      WRITE(MQ,814)
814   FORMAT(1H0,  * IF A RECORD IS NOT USED RECORD ALL RECORD INFORMATI
     1ON ON OUTPUT FILE*)
      GO TO 809
818   CONTINUE
      WRITE(MQ,815)
815   FORMAT(1H0,  * IF A RECORD IS NOT USED RECORD ALL RECORD INFORMATI
     1ON ON TAPE FILE POUT*)
809   CONTINUE
C
      IF(IWTBOP .EQ.1) GO TO 961
      IF(IWTBDT .EQ.1) GO TO 962
      IF(IWTBOP .EQ.2) GO TO 963
      IF(IWTBDT .EQ.2) GO TO 964
      WRITE(MQ,967)
967   FORMAT(1H0,*  DO NOT DESCRIBE RECORDS WITH INVALID TYPE CODES*)
      GO TO 965
961   CONTINUE
      WRITE(MQ,968)
968   FORMAT(1H0, * DESCRIBE ALL RECORDS WITH INVALID TYPE CODES BY FILE
     1 NO, TYPE CODE, FIPS CODE , AND COUNT ON FILE OUTPUT*)
      GO TO 966
962   CONTINUE
      WRITE(MQ,969)
969   FORMAT(1H0,*DESCRIBE ALL DATA FOR RECORDS WITH INVALID TYPE CODE -
     1 IDENTIFY BY NEXT RECORD SET TO BADT, WRITE ON FILE POUT*)
      GO TO 966
963   CONTINUE
      WRITE(MQ,970)
970   FORMAT(1H0, * DESCRIBE ALL RECORDS WITH INVALID TYPE CODES BY FILE
     1 NO, TYPE CODE, FIPS CODE , AND COUNT ON FILE POUT*)
      GO TO 966
964   CONTINUE
      WRITE(MQ,971)
971   FORMAT(1H0,*DESCRIBE ALL DATA FOR RECORDS WITH INVALID TYPE CODE -
     1 IDENTIFY BY NEXT RECORD SET TO BADT, WRITE ON FILE OUTPUT*)
966   CONTINUE
      IF(NWTBD .EQ.0) GO TO 965
      WRITE(MQ,972)
972   FORMAT(1H0, *      THE FOLLOWING LIST ARE EXCEPTIONS- DO NOT DESCRI
     1BE INVALID TYPES IN THE FOLLOWING LIST*,/,11X, * NO,*,5X, *TYPE*,
     2 /)
      DO 973 K = 1,NWTMS
      WRITE(MQ,974) K, IWTBDN(K)
974   FORMAT(1H ,10X, I4,5X,A4)
973   CONTINUE
965   CONTINUE
C
```

```
C
      IF(IFBGEN .EQ. 1) GO TO 819
      IF(IFBGEN .EQ. 2) GO TO 820
      IF(IFBGEN .EQ. 3) GO TO 826
      WRITE(MQ,822)
822   FORMAT(1H0, *DO NOT CHANGE FORMAT CONTROL BITS INPUT*)
      GO TO 821
819   CONTINUE
      WRITE(MQ,823)
823   FORMAT(1H0, *SET FORMAT CONTROL BIT TO SKIP A LINE FOR EACN NEW CO
     1UNTY*)
      GO TO 821
820   CONTINUE
      WRITE(MQ,824)
824   FORMAT(1H0, *SET FORMAT CONTROL BIT TO SKIP A LINE FOR EACN NEW CO
     1UNTY AND TO A NEW PAGE FOR EACH NEW STATE *)
      GO TO 821
826   CONTINUE
      WRITE(MQ,825)
825   FORMAT(1H0,* SET FORMAT CONTROL BIT SO NO LINES ARE SKIPPED*)
821   CONTINUE
C
      IF(ISC .EQ. 1) GO TO 827
      WRITE(MQ,830)
830   FORMAT(1H0,* REQUIRE COMMON PLACE CODE AS WELL AS STATE AND COUNTY
     1 CODE TO DEFINE A COMMON GROUP OF RECORDS*,/,
     2 *    (I.E. JCUFPA(I) AND JCUFPB(I) )*)
      GO TO 828
827   CONTINUE
      WRITE(MQ,829)
829   FORMAT(1H0, *JUST USE STATE AND COUNTY CODES (JCUFPA(I) ) TO DEFIN
     1E A COMMON GROUP OF RECORDS, IGNORE PLACE CODE*)
828   CONTINUE
C
      WRITE(MQ,837)
837   FORMAT(//,10X,*....................*,/,5X,*DESCRIPTION OF INPUT F
     1ILES*,/)
      WRITE(MQ,835)
835   FORMAT(///,1H0, * FILE  RECORD  TYPE  OUTPUT  NO. OF DATA   DESCR
     1IPTION*)
      WRITE(MQ,836)
836   FORMAT(1H , *  NO.    NO.   CODE   ORDER   CHARACTERS*)
      DO 831  I = 1,NMRGT
      WRITE(MQ,834)
834   FORMAT(1H0)
      ITM = NMRGA(I)
      DO 832 J = 1,ITM
      WRITE(MQ,833) I,J,NTYPA(I,J), NLNKL(I,J),NCHRA(I,J),
     1 (NDESCR(I,J,K),K = 1,4)
833   FORMAT(1H , I5,I8,2X,A4,I8,I10,7X,4A10)
832   CONTINUE
831   CONTINUE
      WRITE(MQ,838)
838   FORMAT( /,10X,*....................*,//)
C
C
      ISTRT = 1
      GO TO 130
19    CONTINUE
      ISTRT = 0
C
```

```
C
120     CONTINUE
C       NOW TRANSFER DATA FROM NEXT BUFFER TO CURRENT BUFFER FOR
C       CURRENT FIPS CODE AND SET FLAGS
C
        DO 121 L = 1,NTYPT
        IF(INF(L) .NE.1) GO TO 121
        IF(JNBFPA(L).NE.KCUFPA) GO TO 121
        IF(ISC .EQ.1) GO TO 123
        IF( JNBFPB(L) .NE. KCUFPB) GO TO 121
123     CONTINUE
        JCUFPA(L) = JNBFPA(L)
        JCUFPB(L) = JNBFPB(L)
        JRECTP(L) = JNFCUR(L)
        JNXTP(L) = JNFNXT(L)
        JFBCU(L) = JFBNX(L)
        JRCTCU(L) = JRCTNX(L)
        DO 122 LL = 1,15
        JCURDA(L,LL) = JNXTDA(L,LL)
122     CONTINUE
        ISPFST(L) = JSPFST(L)
        ISPLST(L) = JSPLST(L)
        JSPFST(L) = 0
        JSPLST(L) = 0
        ICF(L) = 1
        INF(L) = 0
C       END OF FILE BUSINESS
        I = NORDI(L)
        IF(IFLEF(I) .EQ. 2) IFLEF(I) = 1
121     CONTINUE
C       CHECK TO SEE IF ALL FILES ARE USED
        DO 124 I = 1,NMRGT
        IF(IFLEF(I) .EQ. 1) GO TO 124
        GO TO 128
124     CONTINUE
        ISTOP = 1
C       SINCE ENTERING 120 MEANS THERE MUST BE VALID DATA IN NEXT BUFFER
C       AND A VALID FIPS CODE THERE IS NOW SOME DATA IN CURRENT BUFFER.
C       PROCESS IT AND EXIT
        GO TO 140
128     CONTINUE
        ICUFPA = KCUFPA
        ICUFPB = KCUFPB
C
C
C
130     CONTINUE
C       NOW READ DATA INTO NEXT BUFFER.
C       IF INF(L) FOR ANY   RECORD OF FILE I IS ONE DONT READ FROM
C       THIS FILE
C
        I = 0
        IRDBK = 1
136     CONTINUE
        I = I + 1
855     CONTINUE
        IF(IFLEF(I) .NE.0) GO TO 131
        NRD = NMRGA(I)
        ITMP = MRD(I)
        INCT = 0
C       DO NOT USE THIS FILE IF ANY INF(I) = 1
```

```
      DO 851 J = 1,NRD
      L = NLNKL(I,J)
      IF(INF(L) .EQ. 1) GO TO 131
851   CONTINUE
      DO 101  J = 1,NRD
      IRDF(J) = 0
101   CONTINUE
      IF(IHR(I) .NE. 1) GO TO 102
C     TAKE FIRST RECORD FROM READ BUFFER
      MFBNX=  JFBHR(I)
      MNFCUR = JHRCUR(I)
      MNFNXT =  JNFHR(I)
      MRCTNX = JRCTHR(I)
      MNBFPA = JHRFPA(I)
      MNBFPB = JHRFPB(I)
      DO 103 K = 1,MXWRD
      MNXDA(K) = JHRDA(I,K)
103   CONTINUE
      KFPNBA(I) = KFPRDA(I)
      KFPNBB(I) = KFPRDB(I)
      IHR(I) = 0
      J = 1
      L = NLNKL(I,1)
      GO TO 135
102   CONTINUE
      J = 0
C
132   CONTINUE
      IRPT = 0
      J = J + 1
      L = NLNKL(I,J)
C
137   CONTINUE
      READ(ITMP,133) MFBNX,MNFCUR,MNFNXT,MRCTNX,MNBFPA,MNBFPB,
     1 (MNXDA(LL) ,LL = 1,MXWRD)
133   FORMAT(A1,A4,1X,A4,I5,1X,A5,A4,1X,11AI0)
C
C     6400 UNIQUE IS 10 CHAR/WRD ASSUMPTION IN FORMAT STATEMENT++++++
C.....NOTICE THIS READ IS NON-ANSI AND RELIES ON A 6400 FTN READ TO CALL
C         FOR MORE INFORMATION THAN IS ON THE RECORD WITHOUT DIRE
C     RESULTS AND WITH THE DATA ON THE RECORD ENTERING VALIDLY
C
C     END OF FILE CHECK, SET INF = 2 IF EOF
C
C     6400 UNIQUE
      IF(EOF(ITMP).EQ.1) 134,135
C     6400 RUN COMPILER BELOW, FTN ABOVE
C     IF(EOF,ITMP) 134,135
C
C
135   CONTINUE
C     NO END OF FILE
      INCT = INCT + 1
C     6400 UNIQUE
C     THIS IS TO CONVERT BLANK-1 TO 0-1
      ITM = MNBFPA .AND. MSKC
      IF(ITM .NE. MSKD) GO TO 147
      MNBFPA =  MSKB .OR. (MNBFPA .AND. MSKA)
147   CONTINUE
:
:
```

```
C     ON FIRST READ PRIME NB CODE ARRAY
      IF(ISTRT .NE. 1) GO TO 854
      IF(INCT .NE.1) GO TO 854
      KFPNBA(I) = MNBFPA
      KFPNBB(I) = MNBFPB
854   CONTINUE
C
C     IF NEW FIPS CODE PREPARE TO EXIT
      IF(KFPNBA(I) .NE. MNBFPA) GO TO 852
      IF (ISC .EQ. 1) GO TO 853
      IF(KFPNBB(I) .NE. MNBFPB) GO TO 852
853   CONTINUE
C
C     TO FIND PROPER J AND L
146   CONTINUE
      IF(MNFCUR .EQ. NTYPIJ(L)) GO TO 139
      J = J + 1
      IF(J .LE. NRD) GO TO 145
C
C     INVALID TYPE CODE SO RECORD NOT USED. RECORD IF NEEDED
      IF(IRPT .NE.1) GO TO 951
      IF(IWTBDP .EQ. 0 .AND. IWTBDT .EQ.0) GO TO 952
      DO 953 K = 1,NWTBD
      IF(MNFCUR .EQ. IWTBDN(K)) GO TO 952
953   CONTINUE
      IF(IWTBDP .EQ.2) GO TO 954
      IF(IWTBDP .NE.1) GO TO 956
      WRITE(MQ,955) I,MNFCUR,MNBFPA,MNBFPB,JRCDCT
955   FORMAT(1H ,*    ON FILE *,I2,* INPUT NOT USED FOR TYPE *,A4,
     1 * IS NOT VALID-- AT CODE *,A5,1X,A4, * AFTER RECORD*,I6)
      GO TO 956
954   CONTINUE
      MNFNXT = 4HBADT
      WRITE(MQ   ,133)MFBNX,MNFCUR,MNFNXT,MRCTNX,MNBFPA,MNBFPB,
     1 (MNXDA(LL) ,LL = 1,MXWRD)
956   CONTINUE
      IF(IWTBDT .EQ.2) GO TO 957
      IF(IWTBDT .NE.1) GO TO 952
      WRITE(MOP , 955) I ,MNFCUR,MNBFPA,MNBFPB,JRCDCT
      GO TO 952
957   CONTINUE
      MNFNXT = 4HBADT
      WRITE(MOP ,133)MFBNX,MNFCUR,MNFNXT,MRCTNX,MNBFPA,MNBFPB,
     1 (MNXDA(LL) ,LL = 1,MXWRD)
952   CONTINUE
      GO TO 132
951   CONTINUE
C
C     TO ALLOW FOR A RECORD OUT OF ORDER
      IRPT = 1
      J = 1
145   CONTINUE
      L = NLNKL(I,J)
      GO TO 146
C
C     J AND L ARE JUST DEFINED
139   CONTINUE
      INF(L) = 1
      NWRD = NCHRCU(L)/NCHRW + 1
      NEXTRA = NCHRCU(L) -(NWRD - 1)*NCHRW
      IF(NEXTRA .EQ.0) NWRD = NWRD - 1
```

16

```
C
C     SPECIAL BUFFER OPERATIONS
      IF(IRDF(J) .EQ. 0) GO TO 161
C     PUT DATA IN SPECIAL BUFFER SINCE MORE THA ONE OF SAME TYPE
C     A SPECIAL BUFFER GROUP HAS THE SAME RECORD TYPE AND FIPS CODE
      NSPNU = NSPNU + 1
      IF(NSPNU .LE. MXSPBU) GO TO 162
      WRITE(MQ,163)  MNBFPA,MNBFPB,MNFCUR
163   FORMAT(///,1H0, *+++++++ERROR STOP, TOO MANY ENTRIES FOR SPECIAL BU
     1FFER++++++*, 3X, *AT FIPS CODE  *,A5,A4, *FOR RECORD TYPE  *,A4)
      STOP 163
162   CONTINUE
C
      IF(JSPFST(L) .EQ.0) JSPFST(L) = NSPNU
      JSPLST(L) = NSPNU
      JFBSP(NSPNU) = MFBNX
      JSPCUR(NSPNU) = MNFCUR
      JNFSP(NSPNU) = MNFNXT
      JRCTSP(NSPNU) = MRCTNX
      JSPFPA(NSPNU) = MNBFPA
      JSPFPB(NSPNU) = MNBFPB
      DO 164 LL = 1,NWRD
      JSPDA(NSPNU,LL) = MNXDA(LL)
164   CONTINUE
      IRPT = 0
C     REFRESH READ BUFFER
      GO TO 137
161   CONTINUE
C
C     NORMAL READ OF DATA INTO NEXT BUFFER
      IRDF(J) = 1
      JFBNX(L)  = MFBNX
      JNFCUR(L) = MNFCUR
      JNFNXT(L) = MNFNXT
      JRCTNX(L) = MRCTNX
      JNBFPA(L) = MNBFPA
      JNBFPB(L) = MNBFPB
      DO 138 LL = 1,NWRD
      JNXTDA(L,LL) = MNXDA(LL)
138   CONTINUE
      GO TO 132
C
C     FILL READ BUFFER AND EXIT TO A NEW FILE
852   CONTINUE
      JFBHR(I) = MFBNX
      JHRCUR(I) = MNFCUR
      JNFHR(I) = MNFNXT
      JRCTHR(I) = MRCTNX
      JHRFPA(I) = MNBFPA
      JHRFPB(I) = MNBFPB
      DO 104  K = 1,MXWRD
      JHRDA(I,K) = MNXDA(K)
104   CONTINUE
      IHR(I) = 1
      KFPRDA(I) = MNBFPA
      KFPRDB(I) = MNBFPB
      GO TO 137
C
C
134   CONTINUE
C     END OF FILE
```

I-198

```
      IFLEF(I) = 2
C     IF NOTHING IN NEXT BUFFER AND NO MORE DATA IN THIS FILE SET
C     THE INDICATOR IFLEF (I) = 1
      DO 943 J = 1,NRD
      L = NLNKL(I,J)
      IF(INF(L) .EQ.1) GO TO 944
943   CONTINUE
      IFLEF(I) = 1
C     CHECK ALL FILES
      DO 945 I = 1,NMRGT
      IF(IFLEF(I) .NE.1) GO TO 944
945   CONTINUE
C     ALL FILES AT EOF AND NO DATA LEFT IN NEXT BUFFER (NOTICE AT EOF
C     THAT THE READ BUFFER MUST BE EMPTY) THEN SET ISTOP = 1, PROCESS
C     CURRENT BUFFER AND EXIT
      ISTOP = 1
      GO TO 140
944   CONTINUE
C
C
131   CONTINUE
C
C     GROUP VALID DATA FOUND IN THIS FIPS READ, REDO FOR ANOTHER FIPS
      DO 947 J = 1,NRD
      L = NLNKL(I,J)
      IF(INF(L) .EQ.1) GO TO 948
947   CONTINUE
      IF(IFLEF(I) .NE.0) GO TO 948
C     NO VALID RECORDS HERE-TRY AGAIN.
      GO TO 855
948   CONTINUE
C
      GO TO(860,861) ,IRDBK
860   CONTINUE
      IF(I .LT. NMRGT) GO TO 136
C
C
C
100   CONTINUE
C     BEGIN NEW CYCLE WITH ALL NEXT BUFFERS FILLED, CURRENT BUFFER
C     FLAGS = 0 AND CURRENT BUFFERS EMPTY OF VALID DATA
C
C
C     FIND CURRENT FIPS CODE
C
      IF(ILEAD .NE.0) GO TO 111
C     USE LOWEST FIPS CODE IN NEXT BUFFER
      KCUFPA = 5H99999
      KCUFPB = 4H9999
      DO 112 L = 1,NTYPT
      IF(INF(L) .NE. 1) GO TO 112
C     NOTICE THAT THIS ASSUMES ALPHABETIC ORDERING IMPLIES NUMERICAL
C     ORDERING
      IF(KCUFPA .GT. JNBFPA(L)) GO TO 113
      IF(ISC .EQ.1) GO TO 112
      IF(KCUFPB .GT. JNBFPB(L)) GO TO 113
      GO TO 112
113   CONTINUE
      KCUFPA = JNBFPA(L)
      KCUFPB = JNBFPB(L)
112   CONTINUE
```

```
C      WITH ILEAD 0 NO RECORDS ARE NOT USED
       GO TO 115
C
111    CONTINUE
C      USE FIPS OF ILEAD FILE
       KCUFPA = KFPNBA(ILEAD)
       KCUFPB = KFPNBB(ILEAD)
C
       IF(IADVL .NE. 1) GO TO 119
C      IF NEXT BUFFER DOES NOT HAVE A ILDR RECORD OF FILE ILEAD,
C         ADVANCE THIS FILE BY GROUPS UNTIL IT DOES
       LL = NLNKL(ILEAD,ILDR)
       IF(INF(L) .EQ.1) GO TO 119
       IF(I+LEF(ILEAD) .EQ. 0) GO TO 857
C      GO TO  1500 AND END IT ALL
       GO TO 1500
857    CONTINUE
       I = ILEAD
C
863    CONTINUE
C      ADVANCE I FILE ONE , RECORD RECORDS BEING DISCARDED
       NRD = NMRGA(I)
       DO 858 J = 1,NRD
       L = NLNKL(I,J)
       IF(INF(L) .NE.1) GO TO 858
       KWTMS = 1
       GO TO 198
C      TO RECORD A RECORD ABOUT TO BE DROPPED
859    CONTINUE
       INF(L) = 0
       IF(JSPFST(L) .EQ.0) GO TO 858
       INU = JSPLST(L) - JSPFST(L) + 1
       DO 869 JJ = 1,INU
       ITMP = JSPFST(L) + JJ - 1
       JFBCU(L) = JFBSP(ITMP)
       JRECTP(L) = JSPCUR(ITMP)
       JNXTP(L) = JNFSP(ITMP)
       JRCTCU(L) = JRCTSP(ITMP)
       JCUFPA(L) = JSPFPA(ITMP)
       JCUFPB(L) = JSPFPB(ITMP)
       DO 176 LL = 1,NWRD
       JCURDA(L,LL) = JSPDA(ITMP,LL)
176    CONTINUE
       KWTMS = 2
       GO TO 198
C      TO RECORD DROPPED FROM SPECIAL BUFFER
870    CONTINUE
869    CONTINUE
       ISTPD = 2
       ICUB = 2
       GO TO 194
C      TO ADJUST BUFFER STORAGE
871    CONTINUE
858    CONTINUE
       IRDBK = 2
       GO TO 855
C      TO READ ONE GROUP FROM I TH FILE
861    CONTINUE
       GOTO(111,116) ,KWTMS
119    CONTINUE
C
```

```
C
      I = 0
C     NOW CHECK POSITIONING OF OTHER FILES
114   CONTINUE
      I = I + 1
      IF(IFLEF(I) .EQ.1) GO TO 116
      IF(I .EQ. ILEAD) GO TO 116
      IF (IADVX .NE. 1) GO TO 862
C     ASSUME NUMERICAL ORDERING ADVANCE TILL MATCH
      IF(KCUFPA .GT. KFPNBA(I)) GO TO 863
      IF(ISC .EQ. 1) GO TO 116
      IF(KCUFPB .GT. KFPNBB(I)) GO TO 863
      GO TO 116
862   CONTINUE
C     NO ORDERING USED, ADVANCE TILL MATCH
      IF (IADVX .NE. 2) GO TO 864
      IF(KCUFPA .NE. KFPNBA(I)) GO TO 863
      IF(ISC .EQ. 1) GO TO 116
      IF(KCUFPB .NE. KFPNBB(I)) GO TO 863
      GO TO 116
864   CONTINUE
C     ADVANCE ONLY IF MATCH IN READ BUFFER
      IF(KCUFPA .NE. KFPRDA(I)) GO TO 116
      IF(ISC .EQ. 1) GO TO 863
      IF(KCUFPB .NE. KFPRDB(I)) GO TO 116
      GO TO 116
116   CONTINUE
      IF (I .LT . NMRGT) GO TO 114
C
115   CONTINUE
C
C
      IF(ISTRT .EQ.1) GO TO 19
C
C
C
C
140   CONTINUE
C     NOW WRITE DATA FROM CURRENT BUFFER ONTO OUTPUT FILE IF BUFFER FLAG
C      IS ONE. IF NO NEXT BUFFER FLAGS ARE + END PROGRAM AFTER WRITE.
C
C     FIRST FIND TYPE OF FIRST 1 FLAGGED RECORD IN NEXT BUFFER
C
      IF(ISTOP .EQ. 1) GO TO 867
      DO 141 L = 1, NTYPT
      IF(INF(L) .NE.1) GO TO 141
      IF( KCUFPA .NE. JNBFPA(L))GO TO 141
      IF(ISC .EQ.1) GO TO 148
      IF(KCUFPB .NE. JNBFPB(L)) GO TO 141
148   CONTINUE
      NTLST = JNFCUR(L)
      GO TO 142
141   CONTINUE
C     ISTOP = 1
867   CONTINUE
      NTLST = NTSTP
142   CONTINUE
C
C     NOW FILL IN NEXT RECORD VALUES
      DO 143 L = 1,NTYPT
      IF(ICF(L) .NE. 1) GO TO 143
      LST = L
      IF(L .EQ. NTYPT) GO TO 143
      ITMP = L + 1
      DO 144 LL = ITMP,NTYPT
      IF(ICF(LL).NE.1) GO TO 144
      JNXTP(L) = JRECTP(LL)
      GO TO 143
144   CONTINUE
143   CONTINUE
C     NOW FILL IN FOR LAST RECORD TO BE WRITTEN FOR THIS TIME
      JNXTP(LST) = NTLST
149   CONTINUE
C
C
C     GENERATION OF FIRST BIT PRINT CONTROL
      IF(IFBGEN .EQ. 0) GO TO 931
C     AND LEAVE FIRST BIT ALONE
      IF(IFBGEN .NE.3) GO TO 932
      DO 933 L = 1,NTYPT
      JFBCU(L) = 1H
C     ALSO FILL SPECIAL BUFFER SPOTS WITH NO SKIP LINE
      IF(ISPFST(L) .EQ.0) GO TO 933
      INU = ISPLST(L) - ISPFST(L) + 1
      DO 937 LL = 1,INU
      LJ = ISPFST(L) + LL - 1
      JFBSP(LJ) = 1H
937   CONTINUE
933   CONTINUE
      GO TO 931
932   CONTINUE
C     PUT BLANK LINE SKIP INTO FIRST RECORD
      IFGO = 0
      DO 934 L = 1,NTYPT
      IF(IFGO .NE.0) GO TO 935
      IF(ICF(L) .NE.1) GO TO 934
      IFGO = 1
      IF(IFBGEN .NE.2) GO TO 936
      ITM = JRECTP(L) .AND. MSKE
      IF(ITM .NE. ISTO) GO TO 936
      ISTO = ITM
      JFBCU(L) = 1H1
      GO TO 938
936   CONTINUE
      JFBCU(L) = 1H0
      GO TO 938
935   CONTINUE
      JFBCU(L) = 1H
938   CONTINUE
C     ALSO FILL SPECIAL BUFFER SPOTS WITH NO SKIP LINE
      IF(ISPFST(L) .EG. 0 ) GO TO 934
      INU = ISPLST(L) - ISPFST(L) + 1
      DO 939 LL = 1,INU
      LJ = ISPFST(L) + LL - 1
      JFBSP(LJ) = 1H
939   CONTINUE
934   CONTINUE
931   CONTINUE
C
C
C
C
150   CONTINUE
```

```
C       NOW WRITE OUTPUT
C
        DO 151 L = 1,NTYPT
        IF(ICF(L) .NE.1) GO TO 152
        NWRD = NCHRCU(L)/NCHRW + 1
        NEXTRA = NCHRCU(L) - (NWRD -1)*NCHRW
        IF(NEXTRA .EQ.0) NWRD = NWRD - 1
        KNXTP = JNXTP(L)
        IF(IVRY .NE.1) GO TO 196
        IF(ISPFST(L) .EQ.0) GO TO 196
        ISTR = JNXTP(L)
        KNXTP = JRECTP(L)
196     CONTINUE
        JRCDCT = JRCDCT + 1
        WRITE(MOT, 156) JFBCU(L), JRECTP(L), KNXTP    ,JRCDCT,JCUFPA(L),
       1 JCUFPB(L), (JCURDA(L,LL),LL= 1,NWRD)
C       6400 UNIQUE IS 10 CHAR/WRD ASSUMPTION IN FORMAT STATEMENT++++++
156     FORMAT(A1,A4,1X,A4,I5,1X,A5,A4,1X,11A10)
C
C       NOW CHECK SPECIAL BUFFER
        IF(IVRY .NE.1) GO TO 157
        IF(ISPFST(L) .EQ.0) GO TO 157
        INU = ISPLST(L) - ISPFST(L) + 1
        DO 171 JJ = 1,INU
        LZ = JJ + ISPFST(L) - 1
        IF(JJ .EQ. INU) GO TO  172
        KNXTP = JRECTP(L)
        GO TO 173
172     CONTINUE
        KNXTP = ISTR
173     CONTINUE
        JRCDCT = JRCDCT + 1
        WRITE(MOT, 156) JFBSP(LZ), JSPCUR(LZ), KNXTP, JRCDCT,JSPFPA(LZ),
       1 JSPFPB(LZ) , (JSPDA(LZ,LL),LL = 1,NWRD)
C
171     CONTINUE
C
C       NOW FLUSH SPECIAL BUFFER OF JUST WRITTEN RECORDS AND ADJUST
C       ACCOUNTS
        ICUB = 1
        ISTPD = 1
        GOTO 194
193     CONTINUE
C
157     CONTINUE
C
        ICF(L) = 0
        GO TO 151
C
C
C       IF REQUESTED RECORD THAT A RECORD IS MISSING FROM NORMAL GROUP
152     CONTINUE
        IF(IWTMS .EQ.0) GO TO 151
        IF(NWTMS .EQ, 0) GO TO 158
        DO 159 LJ = 1,NWTMS
        IF(L .EQ.IWTMSN(LJ))GO TO 151
159     CONTINUE
158     CONTINUE
        IF(IWTMS .EQ.2) GO TO 153
        WRITE(MQ , 154) ICUFPA,ICUFPB,NORDI(L),NORDJ(L),L,NTYPIJ(L),JRCDCT
154     FORMAT(1H ,*NO DATA AT CODE =*, A5,1X,A4 * FILE*,I3,* RECORD*,I3,
      1 * POSITION*,I3,* TYPE *,A4,* AFTER*,I6,* RECORDS*)
       GO TO 151
 153   CONTINUE
       WRITE(MOP, 154) ICUFPA,ICUFPB,NORDI(L),NORDJ(L),L,NTYPIJ(L),JRCDCT
C
C
 151   CONTINUE
C
C
C
C      TEST FOR END OF MERGING
       IF(JRCDCT .GT. NTST) GO TO 1500
       IF(JRCDCT .GT. NTST) GO TO 1500
       IF(ISTOP .NE.1) GO TO 120
 179   CONTINUE
       ENDFILE MOT
       REWIND MOT
       GO TO 1500
C
C
C
C      BEGIN SPECIAL ROUTINE SECTION
C
C
C      STEP DOWN STORAGE IN SPECIAL BUFFER
 194   CONTINUE
       IF(ICUB .EQ.2) GO TO 875
       IBFST = ISPFST(L)
       IBLST = ISPLST(L)
       GO TO 879
 875   CONTINUE
       IBFST = JSPFST(L)
       IBLST = JSPLST(L)
 879   CONTINUE
       IDWN = IBLST - IBFST + 1
       IOV = NSPNU - IBLST
       IF(IOV .EQ.0) GO TO 178
       DO 175 JK = 1,IOV
       ITA = IBFST + JK - 1
       ITB = IBLST + JK
       JFBSP(ITA) = JFBSP(ITB)
       JSPCUR(ITA) = JSPCUR(ITB)
       JRCTSP(ITA) = JRCTSP(ITB)
       JNFSP(ITA) = JNFSP(ITB)
       JSPFPA(ITA) = JSPFPA(ITB)
       JSPFPB(ITA) = JSPFPB(ITB)
       DO 177 LL = 1,MXWRD
       JSPDA(ITA,LL) = JSPDA(ITB,LL)
 177   CONTINUE
 175   CONTINUE
 178   CONTINUE
       DO 876 JK = 1,NTYPT
       IF(JK .NE. L) GO TO 877
       IF(ICUB .EQ.2) GO TO 878
       ISPFST(L) = 0
       ISPLST(L) = 0
       GO TO 87!
 378   CONTINUE
       JSPFST(L) = 0
       JSPLST(L) = 0
 377   CONTINUE
```

```
      IF(ISPFST(JK).LT. IBFST) GO TO 874
      ISPFST(JK)= ISPFST(JK)- IDWN
      ISPLST(JK)= ISPLST(JK)- IDWN
874   CONTINUE
      IF(JSPFST(JK).LT. IBFST) GO TO 876
      JSPFST(JK)= JSPFST(JK)- IDWN
      JSPLST(JK)= JSPLST(JK) - IDWN
876   CONTINUE
      NSPNU = NSPNU - IDWN
      GO TO(198,871),ISTPD
C
C
C     NOW CHECK FOR WRITEING DISCARDED RECORDS
198   CONTINUE
      IF(IWTLSP .EQ. 0) GO TO 126
      IF(IWTLSP.NE.1) GO TO 125
      WRITE(MQ , 118) JNFCUR(L), JNBFPA(L),JNBFPB(L), JRCDCT
118   FORMAT( 1H , * INPUT NOT USED--- TYPE *,A4,  *   CODE *,A5,1X,A4,
     1 * LAST RECORD NO WAS *,I5)
      GO TO 126
125   CONTINUE
      ITMP = 4HDISC
      NWRD = NCHRCU(L)/NCHRW + 1
      NEXTRA = NCHRCU(L) - NWRD*NCHRW
      IF(NEXTRA .EQ.0) NWRD = NWRD - 1
      WRITE(MQ ,156) JFBNX(L),JNFCUR(L),ITMP,JRCDCT,JNBFPA(L),JNBFPB(L),
     1 (JNXTDA(L,LL),LL = 1,NWRD)
126   CONTINUE
      IF(IWTLST .EQ. 0) GO TO 117
      IF(IWTLST .NE.1) GO TO 127
      WRITE(MOP, 118) JNFCUR(L), JNBFPA(L),JNBFPB(L), JRCDCT
      GO TO 117
127   CONTINUE
      ITMP = 4HDISC
      NWRD = NCHRCU(L)/NCHRW + 1
      NEXTRA = NCHRCU(L) - NWRD*NCHRW
      IF(NEXTRA .EQ.0) NWRD = NWRD - 1
      WRITE(MOP,156) JFBNX(L),JNFCUR(L),ITMP,JRCDCT,JNBFPA(L),JNBFPB(L),
     1 (JNXTDA(L,LL),LL = 1,NWRD)
117   CONTINUE
      GO TO(859,870) , KWTMS
C
C
C     END OF FILE MERGE SECTION
C
C
1500  CONTINUE
      RETURN
      END
      SUBROUTINE PUNCH
C
C
C
C     THIS SECTION PUNCHES CARDS
C
C
      COMMON /SPBUF/ JFBSP(300), JSPCUR(300), JNFSP(300),JRCTSP(300),
     1 JSPFPA(300), JSPFPB(300), JSPDA(300,15),NSPNU,
     2 ISPFST(300), ISPLST(300), JSPFST(300), JSPLST(300)
      COMMON / CONS/ IBUGA,IBUGB,IBUGC,IBUGD,IBUGE,IBUGF,IBUGG,IBUGH
     1   , MSKAR(10), MSKA,MSKB,MSKC,MSKD, MSKE,MSKF, MP,MQ,MS,MOT,MOP,
```

24

```
      2 MRD(10),MTA,MTB,MTC,MTD,MTE,MTF,MTG,MTH,MTI,MTJ
      3 ,MXWRD,NTSTP,NCHRW ,MXSPBU
       COMMON /MISC/ KNXDA(15), MNXDA(15), NNXDA(15), ICNT,NTST,ISTOP
      1,JRCDCT
C
C
C
       INOT = 4HPT.8
       NPNCH = 0
       ITMP = MTA
       READ(MP,714) ISC ,IPRT
714    FORMAT(/,3I10)
712    CONTINUE
       READ(MP,711) ITA,ITB,ITC
711    FORMAT(A4,A5,A4)
       IF(ITB .EQ. 5HSTOP )GO TO 713
       NPNCH = NPNCH + 1
       IF(NPNCH  .GT. 300) STOP666
       JSPCUR(NPNCH) = ITA
       JSPFPA(NPNCH) = ITB
       JSPFPB(NPNCH) = ITC
       GO TO 712
713    CONTINUE
C
C
720    CONTINUE
       READ(ITMP,729) MFBNX,MNFCUR,MNFNXT,MRCTNX,MNBFPA,MNBFPB,
      1 (MNXDA(LL) ,LL = 1,MXWRD)
729    FORMAT(A1,A4,1X,A4,I5,1X,A5,A4,1X,11A10)
       IF(EOF(ITMP) .EQ.1) 731,732
C      6400 RUN COMPILER BELOW, FTN ABOVE
C      IF(EOF,ITMP) 731,732
732    CONTINUE
       DO 721 K = 1,NPNCH
       IF(JSPFPA(K) .NE. MNBFPA) GO TO 721
       IF(ISC .EQ.1) GO TO 722
       IF(JSPFPB(K) .NE. MNBFPB) GO TO 721
722    CONTINUE
       IF(JSPCUR(K) .NE. MNFCUR) GO TO 721
       GO TO 723
721    CONTINUE
       GO TO 720
C
723    CONTINUE
C      PUNCH THIS RECORD
       DECODE(10,724,MNXDA(6)) ITA,ITB
724    FORMAT(A4,A6)
       WRITE(MS,726) MFBNX,MNFCUR,MNFNXT,MRCTNX,MNBFPA,MNBFPB,
      1 (MNXDA(I),I=1,5) ,ITA
726    FORMAT(A1,A4,1X,A4,I5,1X,A5,A4,1X,5A10,A4)
       WRITE(MS,727) ITB, (MNXDA(I),I=7,11),INOT,MNFCUR,MNBFPA
727    FORMAT(A6, 5A10, 6X, A4,2X,A4,2X,A5)
       IF(IPRT .NE. 1) GO TO 728
       WRITE(MQ,734) MFBNX,MNFCUR,MNFNXT,MRCTNX,MNBFPA,MNBFPB,
      1 (MNXDA(I),I=1,5) ,ITA
734    FORMAT( 1X,A1,A4,1X,A4,I5,1X,A5,A4,1X,5A10,A4)
       WRITE(MQ,735) ITB, (MNXDA(I),I=7,11),INOT,MNFCUR,MNBFPA
735    FORMAT(1X,A6, 5A10, 6X, A4,2X,A4,2X,A5)
728    CONTINUE
       GO TO 720
731    CONTINUE
```

```
C
C
C     END OF FILE
      REWIND ITMP
C
C
      RETURN
      END
      SUBROUTINE MEET
C
C
C
C     THIS SECTION TO INTRODUCE FILES INTO THE SYSTEM AND PUT INTO
C     STANDARD FORMAT
C     INCLUDE HERE CODING SPECIFIC TO THE FILE TO BE INTRODUCED
      DIMENSION OCMOSE(6,3)
C
C
      COMMON /SPBUF/ JFBSP(300), JSPCUR(300), JNFSP(300),JRCTSP(300),
     1 JSPFPA(300), JSPFPB(300), JSPDA(300,15),NSPNU,
     2 ISPFST(300), ISPLST(300), JSPFST(300), JSPLST(300)
      COMMON / CONS/ IBUGA,IBUGB,IBUGC,IBUGD,IBUGE,IBUGF,IBUGG,IBUGH
     1  , MSKAR(10), MSKA,MSKB,MSKC,MSKD, MSKE,MSKF, MP,MQ,MS,MOT,MOP,
     2 MRD(10),MTA,MTB,MTC,MTD,MTE,MTF,MTG,MTH,MTI,MTJ
     3 ,MXWRD,NTSTP,NCHRW ,MXSPBU
      COMMON /MISC/ KNXDA(15), MNXDA(15), NNXDA(15), ICNT,NTST,ISTOP
     1,JRCDCT
C
C  .........................................................
C
C
C
C     THIS SECTION TAKES A NEW FILE AND PUTS IT INTO STANDARD FORMAT
C
C     VARIOUS TYPES OF INPUT TAPE ARE NUMBERED BY INTRN
C     INTRN = 1 IS ANCET CITY DATA BASE, 1974 POP, DCPACC TAPE NO 5580
C     INTRN = 2 CCBD DATA SELECTION FROM IDA TAPE NO. 669
C     INTRN = 3 F.O. AND MCD BLAST RISK FROM ADAGIO 74 INPUT TAPE
C
C     A GO TO STATEMENT ALLOWS FOR THIS
C     INPUT ON MTA
C
C
C     INITIALIZATION
      READ(MP,201) INTRN
201   FORMAT(/,I10)
C
      GO TO(211,251,271  ,300    ),INTRN
211   CONTINUE
C
C     ANCET INPUT TAPE FORMAT
C     SKIP THREE HEADER RECORDS, 1 SYSTEM, 2 ANCET
      READ(MTA,213)IZIL
      READ(MTA,213)IZIL
      READ(MTA,213)IZIL
213   FORMAT(A1)
      ISTOP = 0
      ISTRT = 1
      IBITF = 1H
      IBITFN = 1H
      ITPN = 4H
      IHSE = 0
      GO TO 220
C
C
C
C     READ INPUT DATA
220   CONTINUE
      GO TO (221         ),INTRN
221   CONTINUE
C
C
C     ANCET INPUT TAPE FORMAT
C     IPOP IS POPULATION PROJECTED TO 1974
C     A IS SEMI MAJOR AXIS, MILES
C     B IS SEMI MINOR AXIS, MILES
C     ALPH IS ANGLE OF ROTATION OF SEMI MAJOR AXIS DEGREES CLOCKWISE
C        FROM THE NORTH
C     IAR IS LAND AREA IN SQ. MILES FROM CCDB
C
C
      READ(MTA,222) NAMA,NAMB,NAMC,ITP,IST,    ICTY,IPLC,FLAD,FLAM,FLAS,
     1 FLOD,FLOM,FLOS
222   FORMAT(2A8 ,A2 ,4X,I2, A2,8X,     4X,A3,A4, 7X,3F3.0,F4.0,2F3.0)
C     6400 UNIQUE
      IF(EOF(MTA ) .EQ.1) 233,234
C     6400 RUN COMPILER BELOW, FTN ABOVE
C     IF(EOF,MTA ) 233,234
233   CONTINUE
      ISTOP = 1
      ITPN = 4HSTOP
      GO TO 240
234   CONTINUE
      READ(MTA,223) IPOP,        A,B,ALPH, CRAD2,IAR
223   FORMAT(I10, 10X    ,2F5.2,F5.1,24X,F9.1,I10)
      FLAD = FLAD + FLAM/60. + FLAS/3600.
      FLOD = FLOD + FLOM/60. + FLOS/3600.
      IF(ITP .NE.12) GO TO 226
      ITPN = 4HUB
      GO TO 225
226   CONTINUE
      IF(ITP .NE.11) GO TO 227
      ITPN = 4HVB
      GO TO 225
227   CONTINUE
      IF(ITP .NE.20) GO TO 228
      ITPN = 4HPB
      GO TO 225
228   CONTINUE
      IF(ITP .NE.21) GO TO 229
      ITPN = 4HLB
      GO TO 225
229   CONTINUE
      IF(ITP .NE.30) GO TO 230
      ITPN = 4HRB
      GO TO 224
230   CONTINUE
      ITPN = 4HHELP
225   CONTINUE
      IAR = 0
      CRAD2 = 0.
224   CONTINUE
```

```
      ITPNN = ITPN
      IF (ISTRT .EQ.0) GO TO 237
      ISTRT = 0
      GO TO 231
237   CONTINUE
C
C
240   CONTINUE
C     WRITE OUTPUT TAPE
      GO TO (241          ),INTRN
241   CONTINUE
      JRCDCT = JRCDCT + 1
C     JUST BELOW IS PROPER WRITE, CHANGED TO SUPRESS 0,S IN WRITE
C     WRITE(MOT, 235) IBITFN,ITPC,ITPN ,JRCDCT,ISTN,ICTYN,IPLCN,NAMAN,
C    1 NAMBN,NAMCN,IPOPN,FLADN,FLODN,AN,BN,ALPHN,IARN,IHSEN,CRAD2N
 235  FORMAT( A1,A4,1X,A4,I5,1X,A2,A3,A4,1X,3A8,I9,2F8.4,2F7.3,F7.2,
     1 I6,I9,F7.3)
C     THIS PUT IN TO SUPRESS 0.
      IF(ITPC .EQ. 4HRB   ) GO TO 242
      WRITE(MOT, 244) IBITFN,ITPC,ITPN ,JRCDCT,ISTN,ICTYN,IPLCN,NAMAN,
     1 NAMBN,NAMCN,IPOPN,FLADN,FLODN,AN,BN,ALPHN,      IHSEN
244   FORMAT( A1,A4,1X,A4,I5,1X,A2,A3,A4,1X,3A8,I9,2F8.4,2F7.3,F7.2,
     1 6X,I9,7X)
      GO TO 243
242   CONTINUE
      WRITE(MOT, 245) IBITFN,ITPC,ITPN ,JRCDCT,ISTN,ICTYN,IPLCN,NAMAN,
     1 NAMBN,NAMCN,IPOPN,FLADN,FLODN,                IARN,IHSEN,CRAD2N
245   FORMAT( A1,A4,1X,A4,I5,1X,A2,A3,A4,1X,3A8,I9,2F8.4,21X,
     1 I6,I9,F7.3)
243   CONTINUE
C     DEBUG HERE()()()()()()()()()()()()()))
      IF(IBUGA .NE.1) GO TO 1510
      WRITE(MQ  ,235) IBITFN,ITPC,ITPN ,JRCDCT,ISTN,ICTYN,IPLCN,NAMAN,
     1 NAMBN,NAMCN,IPOPN,FLADN,FLODN,AN,BN,ALPHN,IARN,IHSEN,CRAD2N
1510  CONTINUE
C     DEBUG HERE()()()()()()()()()()()()()))
      IF(JRCDCT .GT. NTST) GO TO 1500
      IF(ISTOP .NE.1) GO TO  236
      ENDFILE MOT
      REWIND MOT
      GO TO 1500
236   CONTINUE
C
C
231   CONTINUE
      GO TO ( 232         ),INTRN
C     NOW MOVE DATA FROM OLD BUFFER INTO NEW BUFFER
232   CONTINUE
      IBITFN = IBITF
      ITPC = ITPN
      ISTN = IST
      ICTYN = ICTY
      IPLCN = IPLC
      NAMAN = NAMA
      NAMBN = NAMB
      NAMCN = NAMC
      IPOPN = IPOP
      FLADN = FLAD
      FLODN = FLOD
      AN = A
      BN = B
      ALPHN = ALPH
      IARN = IAR
      IHSEN = IHSE
      CRAD2N = CRAD2
      GO TO 221
C
C
C
251   CONTINUE
C     ECONOMIC DATA FROM CITY COUNTY DATA BOOK
C
C
C     KC67-- MEDIAN FAMILIY INCOME DOLLARS,I4
C     KC77-- TOTAL YEAR ROUND HOUSING UNITS, I7
C     KC108-- LOCAL GOVT. DIRECT GENERAL EXP. MILX 10, I5
C     KC121 NO, MANUF. EST., I5
C     KC129-- VALUE ADDED BY MANUFACTURE MILX10, I6
C     KC132-- NO. OF RETAIL ESTAB., I6
C     KC135-- RETAIL SALES, THOUSANDS, I8
C     KC138-- PERCENT RETAIL SALES THAT ARE FOOD SALES, I4
C     KC160-- WHOLESALE SALES, THOUSANDS, I8
C     KC189 VALUE OF FARM SALES, THOUSANDS, I8
C
      IRITFN = 1H
      ITPN = 4HCEAA
      ITPNN = 4HCEAA
      IPLC = 4H
      JRCDCT = 0
256   CONTINUE
      READ(MTA, 252) IST, ICTY,IMARK,KC67,KC77
252   FORMAT(A2,A3,I2,113X,I4,I7)
C     6400 UNIQUE
      IF(EOF(MTA ) .EQ.1) 267,268
C     6400 RUN COMPILER BELOW, FTN ABOVE
C     IF(EOF,MTA ) 267,268
267   CONTINUE
      BACKSPACE MOT
      ITPNN = 4HSTOP
      IST = ISTO
      ICTY = ICTYO
      KC67 = KC670
      KC77 = KC770
      WRITE(MOT,262) IBITFN,ITPN,ITPNN, JRCDCT,IST,ICTY,IPLC,
     1 KC67,KC77,KC108,KC121,KC129,KC132,KC135,KC138,KC160,KC189
      ENDFILE MOT
      REWIND MOT
      GO TO 1500
268   CONTINUE
      ISTO = IST
      ICTYO = ICTY
      KC670 = KC67
      KC770 = KC77
      IF(IMARK .NE.1) STOP 252
      IF(IST .NE. 2H02 .AND. IST .NE. 2H15) GO TO 257
      READ(MTA,213) IZIL
      READ(MTA,213) IZIL
      READ(MTA,213) IZIL
      GO TO 256
257   CONTINUE
      IF(IST .EQ. 2H36 .AND. ICTY .EQ. 3H005) GO TO 263
      IF(IST .EQ. 2H36 .AND. ICTY .EQ. 3H047) GO TO 265
```

```
         IF(IST .EQ. 2H36 .AND. ICTY .EQ. 3H081) GO TO 266
         IF(IST .EQ. 2H36 .AND. ICTY .EQ. 3H085) GO TO 269
         IF(IST .EQ. 2H51 .AND. ICTY .EQ. 3H515) GO TO 246
         IF(IST .EQ. 2H51 .AND. ICTY .EQ. 3H595) GO TO 247
         IF(IST .EQ. 2H51 .AND. ICTY .EQ. 3H775) GO TO 248
         READ(MTA,253) IMARK,KC108,KC121
253      FORMAT(5X,I2,32X,I5,40X,I5)
         IF(IMARK .NE.2) STOP 253
         GO TO 264
263      CONTINUE
         KC108 = 0
         KC121 = 1748
         READ(MTA,213) IZIL
         GO TO 264
265      CONTINUE
         KC108 = 0
         KC121 = 6384
         READ(MTA,213) IZIL
         GO TO 264
266      CONTINUE
         KC108 = 0
         KC121 = 2792
         READ(MTA,213) IZIL
         GO TO 264
269      CONTINUE
         KC108 = 0
         KC121 = 54
         READ(MTA,213) IZIL
         GO TO 264
246      CONTINUE
         KC108 = 0
         KC121 = 60
         READ(MTA,213) IZIL
         GO TO 264
247      CONTINUE
         KC108 = 0
         KC121 = 31
         READ(MTA,213) IZIL
         GO TO 264
248      CONTINUE
         KC108 = 0
         KC121 = 37
         READ(MTA,213) IZIL
         GO TO 264
264      CONTINUE
         READ(MTA, 254) IMARK,KC129,KC132,KC135,KC138,KC160
254      FORMAT(5X,I2,10X,I6,5X,I6,I8,I4,68X,I8)
         IF(IMARK .NE.3) STOP 254
         IF(IST .EQ. 2H29 .AND. ICTY .EQ. 3H510) GO TO 258
         IF(IST .EQ. 2H24 .AND. ICTY .EQ. 3H510) GO TO 258
         READ(MTA, 255) IMARK, KC189
255      FORMAT(5X,I2,24X,I8)
         IF(IMARK .NE.4) STOP 255
         GO TO 259
258      CONTINUE
         KC189 = 0
         READ(MTA,213) IZIL
259      CONTINUE
C
C
         JRCDCT = JRCDCT + 1
```

```
      WRITE(MOT,262) IBITFN,ITPN,ITPNN, JRCDCT,IST,ICTY,IPLC,
     1 KC67,KC77,KC108,KC121,KC129,KC132,KC135,KC138,KC160,KC189
262   FORMAT(A1,A4,1X,A4,I5,1X,A2,A3,A4,1X,
     1 I4,I7,I5,I5,2I6,I8,I4,2I8)
C     DEBUG HERE()()()()()()()()()()()()()))
      IF(IBUGA .NE.1) GO TO 1511
      WRITE(MQ ,262) IBITFN,ITPN,ITPNN, JRCDCT,IST,ICTY,IPLC,
     1 KC67,KC77,KC108,KC121,KC129,KC132,KC135,KC138,KC160,KC189
1511  CONTINUE
C     DEBUG HERE()()()()()()()()()()()()()))
      IF(JRCDCT .GT. NTST) GO TO 1500
      GO TO 256
C
C
C
271   CONTINUE
C     READ BLAST AND FALLOUT RISK DATA FROM ADAGIO INPUT TAPE
C     PUT BLAST RISK ON MOT AND FALLOUT ON MTB
C
      IBITFN = 1H
      ITPB = 4HMRAA
      ITPBN = 4HMRAA
      ITPF = 4HCF
      ITPFN = 4HCF
      IPLC = 4H
C     ***THESE WILL VARY WITH INPUT DATA
      IPOS1 = 2H50
      IPOS2 = 2H75
      IPOS3 = 2H90
      ICDS1 = 1HS
      ICDS2 = 1HW
      JFOCT = 0
      JBLCT = 0
      FLATM = 0.
      FLONM = 0.
      IPOPMU = 0
C
273   CONTINUE
      READ(MTA,272) ITP,LETNX
272   FORMAT( A4,127X,A4)
C     6400 UNIQUE
      IF(EOF(MTA ) .EQ.1) 285,286
C     6400 RUN COMPILER BELOW, FTN ABOVE
C     IF(EOF,MTA ) 285,286
285   CONTINUE
      BACKSPACE MOT
      WRITE(MOT,281) IBITFN,ITPB,ITPBN,JBLCT,IST,ICTY,MCDCD,MNAMA,
     1 MNAMB,MNAMC,MUACIN,IPOPMU,IPOPMR,IPOPMA,FLATM,FLONM,PSIM,
     2 ICLSM,IUACD,DISTM,YLDM
      BACKSPACE MTB
      WRITE(MTB, 277) IBITFN,ITPF,ITPFN,JFOCT, IST,ICTY,IPLC,
     1 ((DCMDSE(IX,JX),JX=1,3),IX=1,2),ICDS1,ICDS2,IPOS1,IPOS2,IPOS3
      ENDFILE MOT
      ENDFILE MTB
      REWIND MOT
      REWIND MTB
      GO TO 1500
286   CONTINUE
280   CONTINUE
      IF(LETNX .EQ.4HCD     ) GO TO 274
      IF(LETNX .EQ.4HDBL  ) GO TO 275
      GO TO 273
C
274   CONTINUE
C     FALLOUT RISK
      READ(MTA,276)ITP,IST,ICTY,((DCMDSE(IX,JX),JX= 1,3),IX= 1,2),LETNX
276   FORMAT(A4,A2,A3,4X,6F10.4,53X,5X,A4)
      JFOCT = JFOCT + 1
      WRITE(MTB, 277) IBITFN,ITPF,ITPFN,JFOCT, IST,ICTY,IPLC,
     1 ((DCMDSF(IX,JX),JX=1,3),IX=1,2),ICDS1,ICDS2,IPOS1,IPOS2,IPOS3
277   FORMAT(A1,A4,1X,A4,I5,1X, A2,A3,A4,1X,6F10.0,40X,2A1,2X,3A2)
C     DEBUG HERE()()()()()()()()()()()()()))(
      IF(IBUGA .NE.1) GO TO 1512
      WRITE(MQ , 277) IBITFN,ITPF,ITPFN,JFOCT, IST,ICTY,IPLC,
     1 ((DCMDSE(IX,JX),JX=1,3),IX=1,2),ICDS1,ICDS2,IPOS1,IPOS2,IPOS3
1512  CONTINUE
C     DEBUG HERE()()()()()()()()()()()()()))(
      IF(JFOCT  .GT. NTST) GO TO 1500
      GO TO 280
C
275   CONTINUE
C     BLAST RISK
C
C
C     IPOPMU URBAN MCD POP
C     IPOPMR-- RURAL MCD POP
C     IPOPMA-- URBANIZED AREAD MCD POP
C     IPOPMT-- TOTAL MCD POP
C     MUACIN-- CODE OF U.A. MCD IS LOCATED IN, IF BLANK NO UA.
C     PSIM-- HIGHEST OVERPRESSURE,PSI
C     DISTM-- DISTANCE FROM WEAPON GIVING HIGHEST OVERPRESSURE,MI.
C     YLDM-- YIELD OF WEAPON GIVING HIGHEST OVERPRESSURE, MT.
C     ICLSM-- CLUSTER NUMBER CONTAINING WEAPON GIVING HIGHEST OVP.
C     IUACD-- CODE OF U.A WITH CLUSTER FOR HIGHEST OVERPRESSURE
C        IF NONE CODE IS BLANKS
C
      READ(MTA,279) ITP,IST,ICTY,MCDCD,MNAMA,MNAMB,MNAMC,YLDM,IUACD,
     1 ICLSM, PSIM, DISTM,MUACIN, IPOPMA,IPOPMT,LETNX
279   FORMAT(A4,A2,A3,A4,      3A8,11X,2X,F4.0,2X,A4,2X,A4,2F8.3,
     1 2X,A4,2X,I7,16X,I8,10X,A4)
C
C     IF NO SEPERATE URBAN AND RURAL POP GIVEN PUT ALL INTO RURAL
      IPOPMR = IPOPMT - IPOPMA
C
      JBLCT = JBLCT + 1
      WRITE(MOT,281) IBITFN,ITPB,ITPBN,JBLCT,IST,ICTY,MCDCD,MNAMA,
     1 MNAMB,MNAMC,MUACIN,IPOPMU,IPOPMR,IPOPMA,FLATM,FLONM,PSIM,
     2 ICLSM,IUACD,DISTM,YLDM
281   FORMAT( A1,A4,1X,A4,I5,1X, A2,A3,A4,1X,3A8,A4,3I7,2F8.3,
     1 F7.2,1X,A4,1X,A4,F6.2,F6.3)
C     DEBUG HERE()()()()()()()()()()()()()))(
      IF(IBUGA .NE.1) GO TO 1513
      WRITE(MQ ,281) IBITFN,ITPB,ITPBN,JBLCT,IST,ICTY,MCDCD,MNAMA,
     1 MNAMB,MNAMC,MUACIN,IPOPMU,IPOPMR,IPORMA,FLATM,FLONM,PSIM.
     2 ICLSM,IUACD,DISTM,YLDM
1513  CONTINUE
C     DEBUG HERE()()()()()()()()()()()()()))(
      GO TO 280
C
C
C
C     TO INTRODUCE MEDLIST RECORDS
```

```
300   CONTINUE
      CALL NOPCHKF(0)
      NTST = 400
      NTST = 999999
      JRCDCT = 0
      ISTOP = 0
      ISTRT = 0
      IFB = 1H
      NTPA = 4HGBBA
      NTPB = 4HGBBB
      NNTPA = NTPA
      NNTPB = NTPB
      NC = 0
      WRITE(MG,303)
303   FORMAT(/////,10X,*LIST OF URBANIZED AREA CODES AND ASSCIATED CODES
     1 FOR PRIME COUNTY FOR FUTURE FIPS ORDERING*  ,///,*  NO. STCNT UAC
     2D       NAME*,/)
302   CONTINUE
      NC = NC + 1
      READ(MP,305) ICT,ISTCO,IUA,NAMEA,NAMEB,NAMEC
305   FORMAT(I4, 1X,A5,1X,A4,5X,3A8)
      ISPFST(NC) = ISTCO
      ISPLST(NC) = IUA
      WRITE(MG,304) ICT, ISTCO, IUA, NAMEA,NAMEB,NAMEC
304   FORMAT(1H , I4,1X,A5, 1X,A4, 5X,3A8)
      IF(IUA .NE. 4H9320) GO TO  302
310   CONTINUE
      READ(MTA, 301) ISTCD,ICOCD,ICNCT, MCDCD,IPLCD,IPLDEC,ICDSM,IUACD,
     1 ICBDCD,NAMEA,NAMEB,NAMEC,ICDTCT,ICDBLG,IEDCD,IURCD,IHSE,IPOP,
     1 ILON,ILAT
301   FORMAT(A2,2X,A3,3X,A1,  A3,A4,A1,3X,A4, A4,15X,A1,3A8,6X,A6,
     1  A1,A5,A1,4X,I7,I8,I10,I10)
      IF(EOF(MTA) .EQ.1) 311,312
311   CONTINUE
      ISTOP = 1
312   CONTINUE
      ICNT = ICNT + 1
      IF(ICNT .GT. NTST) ISTOP = 1
      IF(ISTRT .NE.0) GO TO 313
      ISTRT = 1
      GO TO 315
313   CONTINUE
      IF(ISTOP .NE.1) GO TO 316
      NTPA = 4HSTOP
      NTPB = 4HSTOP
316   CONTINUE
      IF( JUACD .EQ. 4H     ) GO TO 317
      ICDWT = JUACD
      GO TO 319
317   CONTINUE
      IF(JPLCD .EQ. 4H      ) GO TO 318
      ICDWT = JPLCD
      GO TO 319
318   CONTINUE
      ICDWT = 4H
319   CONTINUE
      JRCDCT = JRCDCT + 1
      IF(JRCDCT .EQ.100000) JRCDCT = 0
      WRITE(MOT, 321) IFB, NNTPA,NTPA, JRCDCT , JSTCD,JCOCD, ICDWT,
     1  JPOP,JHSE, FLAT,FLON, JCUTCT, JCDBLG,JEDCD
321   FORMAT( A1, A4,1X,A4,I5,1X, A2,A3,A4,Iv.   I8,I7, 2F8.4,A6,1X,A1,
      1 1X,A5)
       WRITE(MTB, 322) IFB, NNTPB,NTPB, JRCDCT , JSTCD,JCOCD, ICDWT,
      1 JAMEA,JAMEB,JAMEC, JMCDCD, JPLCD, JPLDEC, JCDSM,JCNCT,JCBDCD,
      2 JURCD,JCDTCT,JCDBLG,JEDCD,JHSE,JPOP,FLAT,FLON ,JCNCD
322    FORMAT( A1, A4,1X,A4,I5,1X, A2,A3,A4,IX,   3A8,1X,A3,1X,A4,1X,A1,1
      1X,A4,1X,A1,1X,A1,1X,A1,1X,A6,1X,A1,1X,A5,2I8,2F8.4,1X,A5)
       IF(ISTOP .EQ.1) GO TO 325
315    CONTINUE
       ENCODE(5,326,JSTCO)ISTCD,ICOCD
326    FORMAT(A2,A3)
       DO 327 I = 1,NC
       IF(ISPLST(I) .EQ. IUACD) GO TO 328
327    CONTINUE
       JCNCD = 5H99999
       GO TO 329
328    CONTINUE
       JCNCD = ISPFST(I)
329    CONTINUE
       JSTCD = ISTCD
       JCOCD = ICOCD
       JCNCT = ICNCT
       JMCDCD = MCDCD
       JPLCD = IPLCD
       JPLDEC = IPLDEC
       JCDSM = ICDSM
       JUACD = IUACD
       JCBDCD = ICBDCD
       JAMEA = NAMEA
       JAMEB = NAMEB
       JAMEC = NAMEC
       JCDTCT = ICDTCT
       JCDBLG = ICDBLG
       JEDCD = IEDCD
       JURCD = IURCD
       JHSE = IHSE
       JPOP = IPOP
       JLAT = ILAT
       JLON = ILON
       FLAT = JLAT
       FLON = JLON
       FLAT = FLAT/10000.
       FLON = FLON/10000.
       GO TO 310
325    CONTINUE
       ENDFILE MOT
       ENDFILE MOT
       ENDFILE MTB
       ENDFILE MTB
       RETURN
C
C
1500   CONTINUE
       RETURN
       END
       SUBROUTINE EXTRCT
C
C
C
C      THIS SECTION TO EXTRACT FROM FILES
C      IT IS TO HE WRITTEN
C
```

```
C
      COMMON / CONS/ IBUGA,IBUGB,IBUGC,IBUGD,IBUGE,IBUGF,IBUGG,IBUGH
     1  , MSKAR(10), MSKA,MSKB,MSKC,MSKD, MSKE,MSKF, MP,MQ,MS,MOT,MOP,
     2 MRD(10),MTA,MTB,MTC,MTD,MTE,MTF,MTG,MTH,MTI,MTJ
     3 ,MXWRD,NTSTP,NCHRW ,MXSPBU
      COMMON /MISC/ KNXDA(15), MNXDA(15), NNXDA(15), ICNT,NTST,ISTOP
     1,JRCDCT
C
      A = 1
C
      RETURN
      END
      SUBROUTINE ORDER
C
C
C
      COMMON / CONS/ IBUGA,IBUGB,IBUGC,IBUGD,IBUGE,IBUGF,IBUGG,IBUGH
     1  , MSKAR(10), MSKA,MSKB,MSKC,MSKD, MSKE,MSKF, MP,MQ,MS,MOT,MOP,
     2 MRD(10),MTA,MTB,MTC,MTD,MTE,MTF,MTG,MTH,MTI,MTJ
     3 ,MXWRD,NTSTP,NCHRW ,MXSPBU
      COMMON /MISC/ KNXDA(15), MNXDA(15), NNXDA(15), ICNT,NTST,ISTOP
     1,JRCDCT
C
C
      A = 1
C
C
      RETURN
      END
      SUBROUTINE EDIT
C
C
C
C     ...............................................................
C
C              .......... BEGIN EDITING SECTION ..........
C                         --------------------
C
C
C     EDITING IS CONTROLLED BY INPUT EDIT FILE EITHER INDIVIDUALLY
C     OR WITH COMMON FOR AN ENTIRE STATE( DEFINED BY FIRST 2 CHARACTERS
C     OF FIRST FIPS CODE). INPUT INTO A READ BUFFER M. CONTROL OF
C     DEFINED BY JSPLST(L). AFTER INPUTING A RECORD SEARCH CORRECTION
C     FILE FOR CORRECTIONS. ADVANCE TO N BUFFER READY TO WRITE EXCEPT
C     FOR NEXT RECORD TYPE. ADVANCE TO K BUFFER TO WRITE AFTER PUTTING
C     CURRENT RECORD TYPE IN N BUFFER. WRITE FROM K BUFFER WHEN INPUT
C     OR INSERTS PUSH FROM N BUFFER.
C
C
C
      DIMENSION ICOMDA(20,10),ICOMFS(20),ICOMLS(20),ICOMTP(40)
      DIMENSION ISTLST(50),ISTTP(50),IPLLST(100),IPLTP(100),IPLFLG(100)
      COMMON / FILOR/ NMRGA(10), NTYPA(10,10) , NORDI(100), NORDJ(100),
     1 NTYPIJ(100), NLNKL(10,10), MDESCA(100   ,4)   , NCHRA(10,10),
     2   NCHRCU(100),NMRGT,NTYPT
      COMMON / FNBUF/ JFBCU(100),JRECTP(100), JNXTP(100), JRCTCU(100),
     1 JCUFPA(100), JCUFPB(100),JSPNFA(300),JSPNFB(300),JSPNRT(300),
     2 JSPXTA(600), ICF(100)
     3, JFBNX(100), JNFCUR(100),JNFNXT(100), JRCTNX(100), JNBFPA(100),
     4 JNBFPB(100),JNXTDA(100,15),INF(100)
      COMMON /SPBUF/ JFBSP(300), JSPCUR(300), JNFSP(300),JRCTSP(300),
     1 JSPFPA(300), JSPFPB(300), JSPDA(300,15),NSPNU,
     2 ISPFST(300), ISPLST(300), JSPFST(300), JSPLST(300)
      COMMON / CONS/ IBUGA,IBUGB,IBUGC,IBUGD,IBUGE,IBUGF,IBUGG,IBUGH
     1  , MSKAR(10), MSKA,MSKB,MSKC,MSKD, MSKE,MSKF, MP,MQ,MS,MOT,MOP,
     2 MRD(10),MTA,MTB,MTC,MTD,MTE,MTF,MTG,MTH,MTI,MTJ
     3 ,MXWRD,NTSTP,NCHRW ,MXSPBU
      COMMON /MISC/ KNXDA(15), MNXDA(15), NNXDA(15), ICNT,NTST,ISTOP
     1,JRCDCT
C
C
C
C     INITIALIZE
      NTST = 9999999
      JDOC = 0
0     JDOD = 0
      JDOF = 0
      JDOG = 0
      NIST = 0
      MBUFFL = 0
      NBUFFL = 0
      KBUFFL = 0
      DO 2 L = 1,100
      JNBFPA(L) = 5H99999
      JNBFPB(L) = 4H9999
2     CONTINUE
C
C
      WRITE(MO,301)
301   FORMAT( ///,30X,*......... EDITING OF INPUT FILE .........*,///)
```

```
      READ(MP,311)  NTYPT,ISC,IPEDT
311   FORMAT(/,5I10)
C     READ ORDER FUR INSERTING
      DO 315 I = 1,NTYPT
      READ(MP,316) L,ITA,ITB,ITC,ITD,ITE,ITF
316   FORMAT( I3,3X,A4,I5,5X,4A10)
      NTYPIJ(L) = ITA
      NCHRCU(L) = ITB
      MDESCR(L,1) = ITC
      MDESCR(L,2) = ITD
      MDESCR(L,3) = ITE
      MDESCR(L,4) = ITF
315   CONTINUE
      WRITE(MO,312)
312   FORMAT( //,1H0,*ASSUMED RECORD ORDER ON FILE TO EDIT IS AS FOLLOWS
     1 *,/,*  NO.  TYPE  NO. CHAR.       DESCRIPTION*,/)
      DO 313 L = 1,NTYPT
      WRITE(MO,314) L,NTYPIJ(L),NCHRCU(L),(MDESCR(L,I),I=1,4)
314   FORMAT(1H , I4,2X,A4,6X,I6,2X,4A10)
313   CONTINUE
C
C     READ COMMON CHANGE ITEMS
      ICOMNU = 0
306   CONTINUE
      ICOMNU = ICOMNU + 1
      READ(MP,307) ICOMFS(ICOMNU),ICOMLS(ICOMNU),(ICOMDA(ICOMNU,LL),
     1 LL = 1,7)
307   FORMAT(I3,2X,I3,2X,7A10)
      IF(ICOMFS(ICOMNU) .EQ.999) GO TO 903
      IF(ICOMNU .NE.1) GO TO 904
      WRITE(MO,905)
905   FORMAT(1H0,20X,*DEFINITION OF COMMON CHANGE TYPES*,/,
     1 *  NO. START  LAST  INSERT TEXT*,/ )
904   CONTINUE
      WRITE(MO,906) ICOMNU,ICOMFS(ICOMNU),ICOMLS(ICOMNU),(ICOMDA(ICOMNU,
     1 LL),LL = 1,7)
906   FORMAT(1H ,I4,2I6,4X,7A10)
      GO TO 306
903   CONTINUE
      ICOMNU = ICOMNU - 1
C
C     READ STATE EUITING ITEMS
      NST = 0
318   CONTINUE
      READ(MP,319)ITA,ITB
319   FORMAT( A2, 6X,I2)
      IF(ITA .EQ. 2H99 ) GO TO 320
      ITA = ITA .AND. MSKE
      NST = NST + 1
      IF(NST .NE.1 .OR. IPEDT .NE.1) GO TO 908
      WRITE(MO,909)
909   FORMAT(//,1H0,20X,*LIST OF COMMON STATE EDITING ITEMS*,//)
908   CONTINUE
      ISTLST(NST) = ITA
      ISTTP(NST) = ITB
      IF(IPEDT  .NE.1) GO TO 881
      WRITE(MO,882) NST, ISTLST(NST),ISTTP(NST)
882   FORMAT(1H0, *  FOR *, I3, * TH ITEM FOR ALL RECORDS WITH STATE NUM
     1BER *, A2 ,* TAKE ACTION NUMBFR*,I3)
881   CONTINUE
      GO TO 318
```

```
320   CONTINUF
C
C     READ PLACE EUITING ITEMS
      NPL = 0
      MSKG = 77777777000000000000B
911   CONTINUF
      READ(MP,912) ITA,ITB,ITC
912   FORMAT(A4,I2,I2)
      IF(ITA .EQ. 4H9999) GO TO 913
      ITA = ITA .AND. MSKG
      NPL = NPL + 1
      IF(NPL .NE.1 .OR. IPEDT .NE.1) GO TO  914
      WRITE(MO,915)
915   FORMAT(//,1H0,20X,*LIST OF COMMON PLACE EDITING ITEMS*,//)
914   CONTINUE
      IPLLST(NPL) = ITA
      IPLTP(NPL) = ITB
      IPLFLG(NPL) = ITC
      IF(IPEDT .NE.1) GO TO 916
      WRITE(MO,917) NPL, IPLLST(NPL),IPLTP(NPL),IPLFLG(NPL)
917   FORMAT(1H0, *  FOR *, I3, * TH ITEM FOR ALL RECORDS WITH PLACE NUM
     1BER *, A4 ,* TAKE ACTION NUMBER*,I3,* WITH REPEAT FLAG =*,I2)
916   CONTINUF
      GO TO 911
913   CONTINUF
C
C     READ INDIVIDUAL EDITING ITEMS
      NSPNU = 0
321   CONTINUF
      READ(MP,322) MFBNX,MNFCUR,MNFNXT,MRCTNX,MNBFPA,MNBFPB,
     1(MNXDA(I),I= 1,5),ITA
322   FORMAT(A1,A4,1X,A4,I5,1X,A5,A4,1X,5A10,A4)
      IF(MNFCUR .EQ. 4HSTOP  ) GO TO 324
      NSPNU = NSPNU + 1
      IF(NSPNU .LE. 300) GO TO 880
      WRITE(MO, 885) MNFCUR, MNBFPA
885   FORMAT(1H0,* TOO MANY INPUT CORRECTIONS AT RECORD TYPE *,A4,
     1 * AND CODE *,A5, *---STOP---*)
      STOP 322
880   CONTINUF
      IF(IPEDT .NE.1) GO TO 883
      IF(NSPNU .NE. 1) GO TO 883
      WRITE(MO,884)
884   FORMAT(///,1H0,* ... LIST OF INDIVIDUAL EDIT CHANGES ...*,///)
883   CONTINUF
      READ(MP,323) ITB,(MNXDA(I),I=7,11) ,IFLG ,ITC,ITD,ITE
323   FORMAT( A6,5A10 ,1X,I3, 1X,A5,A4,1X,A4)
      ENCODE( 10,325,MNXDA(6)) ITA,ITB
325   FORMAT(A4,A6)
      JFBSP(NSPNU) = MFBNX
      JSPCUR(NSPNU) = MNFCUR
      JNFSP(NSPNU) = MNFNXT
      JRCTSP(NSPNU) = MRCTNX
      JSPFPA(NSPNU) = MNBFPA
      JSPFPB(NSPNU) = MNBFPB
      DO 326 LL = 1,15
      JSPDA(NSPNU,LL) = MNXDA(LL)
326   CONTINUF
      JSPLST(NSPNU) = IFLG
      JSPNFA(NSPNU) = ITC
      JSPNFB(NSPNU) = ITD
      JSPNRT(NSPNU) = ITE
      DO 329 LL = 1,NTYPT
      IF(MNFCUR.EQ.NTYPIJ(LL)) GO TO 892
329   CONTINUE
      ISPLST(NSPNU) = 99
      WRITE(MO,895) NSPNU,MNFCUR,MNBFPA
895   FORMAT(1H0 ,*...AT INPUT NO *,I3,* RECORD TYPE *,A4, * WITH CODE *
     1 ,A5,  * IS INVALID TYPE*)
      GO TO 893
892   CONTINUE
      ISPLST(NSPNU) = LL
893   CONTINUE
      IF(IPEDT .NE. 1) GO TO 327
      WRITE(MO,898) IFLG,ITE,ITC,ITD
898   FORMAT(1H0,* ACTION TYPE*,I3,* AUXILARY RECORD TYPE AND CODE ARE *
     1 , A4,3X,A5,A4)
      WRITE(MO , 328) MFBNX ,MNFCUR,MNFNXT,MRCTNX,MNBFPA,MNBFPB,
     1 (MNXDA(LL),LL= 1,11)
328   FORMAT(1H ,A1,A4,1X,A4,I5,1X,A5,A4,1X,10A10,A9)
327   CONTINUE
      GO TO 321
324   CONTINUE
C
C
C
C     IADJ = 0 INDICATES NO INTERFERENCE WITH SEQUENCE, NEXT RECORD
C     TYPE O.K., IF1 MUST BE ADJUSTED BEFORE WRITE
C     JSPLST(L) IS ACTION TYPE
C     JSPFST(L) IF ONE THIS CORRECTION HAS BEEN USED
C     ISPFST(L) = 1 INSERT AFTER NEXT WRITE(N BUFFER)
C     ISPFST(L) = 2 INSERT AFTER TWO WRITES (AFTER M BUFFER WRITE)
C     ISPLST(L) PLACE OF RECORD IN LIST
C     M BUFFER JUST READ DATA
C     N BUFFER READY TO WRITE EXCEPT FOR NEXR FIPS CODE
C     K BUFFER WRITE FROM HERE
C
C
C
330   CONTINUE
      IF(MBUFFL .EQ.1)GO TO 350
      ICNT = ICNT + 1
      IF(ICNT .GT. NTST) RETURN
      READ(MTA ,331) MFBNX,MNFCUR,MNFNXT,MRCTNX,MNBFPA,MNBFPB,
     1 (MNXDA(LL) ,LL = 1,MXWRD)
331   FORMAT(A1,A4,1X,A4,I5,1X,A5,A4,1X,11A10)
C     6400 UNIQUE
      IF(EOF(MTA ).EQ.1) 333,332
C     6400 RUN COMPILER BELOW, FTN ABOVE
C     IF(EOF,MTA ) 333,332
333   CONTINUE
C     END OF FILE
      ISTOP = 1
      MBUFFL = 0
      GO TO 360
332   CONTINUE
C     NO END OF FILE
      MBUFFL = 1
C     6400 UNIQUE
C     THIS IS TO CONVERT BLANK-1 TO 0-1
      ITM = MNBFPA .AND. MSKC
      IF(ITM .NE. MSKD) GO TO 334
      MNBFPA = MSKB .OR. (MNBFPA .AND. MSKA)
334   CONTINUF
C
      IF(NST. EQ. 0) GO TO 338
C     6400 UNIQUE
      ITMP =  MNBFPA .AND. MSKE
      DO 339 I = 1,NST
      IF(ISTLST(I) .NE.ITMP) GO TO 339
      IF(ISTTP(I) .NE. 1) GO TO 887
      MBUFFL = 0
      GO TO 330
887   CONTINUF
      IF(ISTTP(I) .LE. 10) GO TO  886
      JDOG = JDOG + 1
      ICOMTP(JDOG) = ISTTP(I) - 10
886   CONTINUF
C     OTHER ACTIONS DO NOT MAKE SENSE FOR A WHOLE STATE
339   CONTINUF
338   CONTINUF
C
      IF(NPL .EQ.0) GO TO 920
C     6400 UNIQUE
      ITMP = MNBFPB .AND. MSKG
      DO 921 I = 1,NPL
      IF(IPLLST(I) .NE. ITMP) GO TO 921
      IF(IPLFLG(I) .EQ.2) GO TO 921
C     ONLY DO ONCE IF PLACE FLAG IS 0
      IF(IPLFLG(I) .EQ.0) IPLFLG(I) = 2
      IF(IPLTP(I) .NE.1) GO TO 923
      MBUFFL = 0
      GO TO 330
923   CONTINUF
      IF(IPLTP(I) .LE. 10)  GO TO 924
      JDOG = JDOG + 1
      ICOMTP(JDOG) = IPLTP(I) - 10
924   CONTINUF
921   CONTINUF
920   CONTINUF
C
C    SEARCH SPECIAL EDITING INSTRUCTIONS
      IF(NSPNU .EQ.0) GO TO 343
      DO 335 L = 1,NSPNU
      IF(JSPFST(L) .EQ.1) GO TO 335
      IF(JSPFPA(L) .NE.MNBFPA) GO TO 335
      IF(ISC .EQ.1) GO TO 336
      IF(JSPFPB(L) .NE. MNBFPB) GO TO 335
336   CONTINUE
      IF(JSPCUR(L) .NE. MNFCUR) GO TO 335
      IF(JSPLST(L) .NE.1) GO TO  888
      JSPFST(L) = 1
      JDOG = 0
      MBUFFL = 0
      GO TO 330
888   CONTINUE
      IF(JSPLST(L) .NE.4) GO TO 889
      JDOD = JDOD + 1
      JSPFST(L) = 1
      ISPFST(L) = 2
      JSPFPA(L) = JSPNFA(L)
      JSPFPB(L) = JSPNFB(L)
      JSPCUR(L) = JSPNRT(L)
      GO TO 335
889   CONTINUE
      IF(JSPLST(L) .NE.5) GO TO  890
      JSPFST(L) = 1
      NIST = NIST + 1
      ISPFST(L) = 1
      JSPFPA(L) = JSPNFA(L)
      JSPFPB(L) = JSPNFB(L)
      JSPCUR(L) = JSPNRT(L)
      GO TO 335
890   CONTINUE
      IF(JSPLST(L).NE.6) GO TO 896
C     CONVERT MOVE TO INSERT ORDER
      JSPFST(L) = 0
      JSPLST(L) = 2
      JSPFPA(L) = JSPNFA(L)
      JSPFPB(L) = JSPNFB(L)
      JSPCUR(L) = JSPNRT(L)
      JFBSP(L) = MFBNX
      JNFSP(L) = MNFNXT
      DO 894 LL = 1,15
      JSPDA(L,LL) = MNXDA(LL)
894   CONTINUE
      MBUFFL = 0
      GO TO 330
896   CONTINUE
      IF(JSPLST(L) .LE.10) GO TO 891
      JSPFST(L) = 1
      JDOG = JDOG + 1
      ICOMTP(JDOG) = JSPLST(L) - 10
      GO TO 335
891   CONTINUE
      IF(JSPLST(L) .EQ.3) JDOC = L
335   CONTINUE
343   CONTINUE
C
C     FIND PLACE OF THIS RECORD TYPE
      DO 341 LL = 1,NTYPT
      IF(MNFCUR .NE. NTYPIJ(LL)) GO TO 341
      LLINM = LL
      GO TO 897
341   CONTINUE
897   CONTINUE
C
C     LOOK FOR INSERTS
      IF(NSPNU .EQ.0) GO TO 344
      DO 337 L = 1,NSPNU
      IF(JSPLST(L) .NE. 2) GO TO 337
      IF(JSPFST(L) .EQ.1) GO TO 337
      IF(JSPFPA(L) .GT. MNBFPA) GO TO 337
      IF(JSPFPA(L) .LT. MNBFPA) GO TO 342
      IF(ISC .EQ.1) GO TO 340
      IF(JSPFPB(L) .GT. MNBFPB) GO TO 337
      IF(JSPFPB(L) .LT. MNBFPB) GO TO 342
340   CONTINUE
C     NOW CHECK RECORD ORDER
      IF(ISPLST(L) .GT. LLINM) GO TO 337
342   CONTINUE
      NIST = NIST + 1
      ISPFST(L) = 1
      JSPFST(L) = 1
```

```
337    CONTINUE
344    CONTINUE
C
C
C
350    CONTINUE
C      NOW INTERPET ACTION
C      MAKE CHANGES
       IF(JDOC .EQ.0) GO TO 351
       MFBNX = JFBSP(JDOC)
       MNFCUR = JSPCUR(JDOC)
       MNFNXT = JNFSP(JDOC)
       MRCTNX = JRCTSP(JDOC)
       MNBFPA = JSPFPA(JDOC)
       MNBFPB = JSPFPB(JDOC)
       DO 352 LL = 1,15
       MNXDA(LL) = JSPDA(JDOC,LL)
352    CONTINUE
       JDOC = 0
351    CONTINUE
C
       IF (JDOC .EQ. 0) GO TO 353
C      IF THE TYPE IS GREATER THAN 20 THE USER MUST INSERT SPECIAL CODE
C      TO MAKE THE DESIRED TYPE OF CHANGES
       DO 1001 K = 1,JDOC
       IT = ICOMTP(K)
       IF (IT .GT. 10) GO TO 1050
       IF(IT .LE. ICOMNU)GO TO 1002
       WRITE(MO,1003) MNFNXT, MNBFPA,MNBFPB, IT
1003   FORMAT(1H0, *---TROUBLE AT RECORD TYPE *,A4,* AND CODE *,A5,A4,*
      1 FOR CHANGE TYPE IS *,I4,* WHICH IS MORE THAN INPUT NO.*)
       STOP 1002
1002   CONTINUE
       ITF = ICOMFS(IT)
       ITL = ICOMLS(IT)
       IWDF = (ITF - 1)/NCHRW + 1
       ITA = ITF -(IWDF - 1)*NCHRW
       ITSHF =(NCHRW + 1 - ITA)*6
       ICHF = (IWDF - 1)*NCHRW + 1
       JWDF = IWDF
       JWDDA = 1
       JDACT = 1
       JFCT = 1
1005   CONTINUE
       IAR = NCHRW + 1 - JFCT
       IF(ICHF .GE. ITF. AND. ICHF .LE. ITL) GO TO 1006
       JWDCN = (MNXDA(JWDF) .AND. MSKAR(IAR)).OR. (JWDCN.AND..NOT.MSKAR
      1(IAR))
       GO TO 1010
1006   CONTINUE
       IARDA = NCHRW + 1 - JDACT
       ITMP = ICOMDA(JWDDA).AND. MSKAR(IARDA)
C      SHIFT(A,B) IS A 6400 ROUTINE WHICH SHIFTS A LEFT CIRCULARLY B BITS
C      CIRCULAR SHIFT IS ESSENTIAL HERE
       ITM = SHIFT (ITMP,ITSHF)
       JWDCN = (ITM .AND. MSKAR(IAR)).OR. (JWDCN. AND..NOT.MSKAR(IAR))
       JDACT = JDACT + 1
       IF(JDACT .LE. NCHRW) GO TO 1010
       JDACT = 1
       JWDDA = JWDDA + 1
1010   CONTINUE
```

```
C****BUG START*********
      IF(IBUGC .NE.1) GO TO 901
      WRITE(MO,902) ITF,ITL,IWDF,ITSHF,JWDF,JWDDA.      JDACT,JFCT,
     1 IAR,IARDA, ICOMDA(JWDDA),ITMP,ITM,JWDCN, ICHF,JRCDCT
902   FORMAT( 1H ,* BUG CONSTRUCTING*, 10I4, A10, 3A10, 2I4)
901   CONTINUF
C*******BUG END*******
      JFCT = JFCT + 1
      ICHF = ICHF + 1
      IF(JFCT .LE. NCHRW) GO TO 1005
      JFCT = 1
      MNXDA(JWDF) = JWDCN
      JWDF = JWDF + 1
      IF( ICHF .LE. ITL) GO TO 1005
1001  CONTINUF
C
1050  CONTINUE
C
C     USER INSERT SPECIAL CODE HERE FOR UNIQUE COMMON CHANGES
C     STATEMENT NUMBERS FROM 1051 TO 1099 MAY BE USED
C
      JDOG = 0
353   CONTINUF
C
C
C
360   CONTINUF
C     WRITE RECORD FROM K BUFFER
C****BUGG START***********
      IF(IBUGA .NE. 1) GO TO 376
      WRITE(MO, 377) KBUFFL,NBUFFL.MBUFFL,NIST,JDOC,JDOD,JDOF,JDOG,LU
377   FORMAT(1H0, * BUGG-- A-- K N M BF FL,NIST JDO C D F G LU =*,9I4,
     1/ ,*  CONT OF  K,N,M BUF =*)
      WRITE(MO,388)KFBNX,KNFCUR,KNFNXT,KRCTNX,KNBFPA,KNBFPB,
     1 (KNXDA(LL),LL= 1,11 )
388   FORMAT(1H ,A1,A4,1X,A4,I5,1X.A5,A4,1X,10A10,A9)
      WRITE(MO,388)NFBNX,NNFCUR,NNFNXT,NRCTNX,NNBFPA,NNBFPB,
     1 (NNXDA(LL),LL= 1,11 )
      WRITE(MO,388)MFBNX,MNFCUR,MNFNXT,MRCTNX,MNBFPA,MNBFPB,
     1 (MNXDA(LL),LL= 1,11 )
      WRITE(MO,389) LLINK,LLINN,LLINM
389   FORMAT(1H ,*  LLINK-LLINN- LLINM = *,3I8)
      DO 378 LL = 1,NSPNU
      WRITE(MO,379)  LL,ISPFST(LL),ISPLST(LL),JSPFST(LL),JSPLST(LL),
     1 JSPCUR(LL),JSPFPA(LL),JSPFPB(LL)  ,JSPNFA(LL),JSPNFB(LL),JSPNRT(LL
     2)
379   FORMAT(1H ,* L=*,I3,*  I J SP FST LST = *,4I6,* TYPE FIPS SPEC=*,
     1 2X, A4,1X,A5,A4,2X, A5,A4,1X,A4)
378   CONTINUF
376   CONTINUE
C******BUG END *******
      IF(KBUFFL .NE. 1) GO TO 362
      IF(NBUFFL .EQ.1) GO TO 363
      IF(ISTOP .EQ. 1) GO TO 363
      GO TO 370
363   CONTINUF
      NWRD = NCHRCU(LLINK)/NCHRW + 1
      NEXTRA = NCHRCU(LLINK) - (NWRD-1)*NCHRW
      IF(NEXTRA .EQ.0) NWRD = NWRD - 1
      JRCDCT = JRCDCT + 1
      ITA = NNFCUR
      WRITE(MOT, 361) KFBNX, KNFCUR,ITA   ,JRCDCT,KNBFPA,KNBFPB,
     1(KNXDA(LL),LL= 1,NWRD)
361   FORMAT(A1,A4,1X,A4,I5,1X,A5,A4,1X,11A10)
C****BUGG STAPT***********
      IF(IBUGR .NE. 1) GO TO 372
      WRITE(MO , 361) KFBNX, KNFCUR,ITA   ,JRCDCT,KNBFPA,KNBFPB,
     1(KNXDA(LL),LL= 1,NWRD)
372   CONTINUF
C*****BUG END *******
      KBUFFL = 0
362   CONTINUE
      IF(KBUFFL .EQ.1) GO TO 360
      IF(NBUFFL .NE.1) GO TO 364
      KFBNX = NFBNX
      KNFCUR = NNFCUR
      KNFNXT = NNFNXT
      KRCTNX = NRCTNX
      KNBFPA = NNBFPA
      KNBFPB = NNBFPB
      DO 365 LL = 1,15
      KNXDA(LL) = NNXDA(LL)
365   CONTINUF
      LLINK = LLINN
      NBUFFL = 0
      KBUFFL = 1
364   CONTINUF
C
C
C
370   CONTINUF
C     CHECK FOR INSERTS
      IF(NIST .EQ.0) GO TO 380
C     WRITE INSERTS
C     FIND SMALLEST
      MFIPVA = 5H99999
      MFIPVB = 4H9999
      LTYP = 9999
      DO 373 L = 1, NSPNU
      IF(ISPFST(L) .NE.1) GO TO 373
      IF(JSPFPA(L) .GT. MFIPVA) GO TO 373
      IF(ISC .EQ. 1 )GO TO 374
      IF(JSPFPB(L) .GT. MFIPVB) GO TO 373
374   CONTINUF
      MFIPVA = JSPFPA(L)
      MFIPVB = JSPFPB(L)
      IF(ISPLST(L) .GT. LTYP) GO TO 373
      LTYP = ISPLST(L)
      LU = L
373   CONTINUF
      IF(NBUFFL .EQ.1) GO TO 360
      NFBNX = JFBSP(LU)
      NNFCUR = JSPCUR(LU)
      NNFNXT = JNFSP(LU)
      NRCTNX = JRCTSP(LU)
      NNBFPA = JSPFPA(LU)
      NNBFPB = JSPFPB(LU)
      DO 375 LL = 1,15
      NNXDA(LL) = JSPDA(LU,LL)
375   CONTINUF
      LLINN = ISPLST(LU)
      JSPFST(LU) = 1
      ISPFST(LU) = 0
      NBUFFL = 1
      NIST = NIST - 1
      GO TO 360
C
C
C
380   CONTINUF
C     ADVANCE FROM M BUFFER TO N BUFFER
      IF(ISTOP .NE.1) GO TO 381
C     CHECK FOR INSERTS AFTER LAST INPUT RECORD
C     INSERTS PUT IN IN ORDER THEY APPEAR IN THE SPECIAL BUFFER
      IF(NSPNU .EQ.0) GO TO 387
      DO 371 L = 1,NSPNU
      IF(JSPLST(L) .NE.2) GO TO 371
      IF(JSPFST(L) .EQ.1) GO TO 371
      IF(JSPFPA(L) .LT. KNBFPA) GO TO 371
      IF(JSPFPA(L) .GT. KNBFPA) GO TO 368
      IF(ISC .EQ.1) GO TO 366
      IF(JSPFPB(L) .LT. KNBFPB) GO TO 371
      IF(JSPFPB(L) .GT. KNBFPB) GO TO 368
366   CONTINUF
      IF(ISPLST(L) .LT. LLINK) GO TO 371
368   CONTINUF
      NFBNX = JFBSP(L)
      NNFCUR = JSPCUR(L)
      NNFNXT = JNFSP(L)
      NRCTNX = JRCTSP(L)
      NNBFPA = JSPFPA(L)
      NNBFPB = JSPFPB(L)
      DO 367 LL = 1,15
      NNXDA(LL) = JSPDA(L,LL)
367   CONTINUF
      JSPFST(L) = 1
      NBUFFL = 1
      GO TO 385
371   CONTINUF
387   CONTINUF
      IF(NBUFFL .NE. 0. .OR. KBUFFL .NE. 0) GO TO 385
      ENDFILE MOT
      ENDFILE MOT
      REWIND MOT
      RETURN
385   CONTINUF
C     TO FINISH SINCF ALL DATA HAS BEEN USED
      IF(NBUFFL .EQ.1) GO TO 360
      NNFCUR = 4HSTOP
      GO TO 360
381   CONTINUF
      IF(NBUFFL .EQ.1) GO TO 360
      IF(MBUFFL .EQ.0) GO TO 330
      NFBNX = MFBNX
      NNFCUR = MNFCUR
      NNFNXT = MNFNXT
      NRCTNX = MRCTNX
      NNBFPA = MNBFPA
      NNBFPB = MNBFPB
      DO 382 LL = 1,15
      NNXDA(LL) = MNXDA(LL)
382   CONTINUF
      IF(JDOD .EQ.0) GO TO 383
      DO 384 L = 1,NSPNU
      IF(ISPFST(L) .NE.2) GO TO 386
      ISPFST(L) = 1
      NIST = NIST + 1
386   CONTINUE
384   CONTINUE
      JDOD = 0
383   CONTINUE
      LLINN = LLINM
      NBUFFL = 1
      MBUFFL = 0
      GO TO 330
C
C
C
C
C
      END
```

COMPUTER PROGRAM DESCRIPTION

NAME: SNORT

SYNOPSIS: Simple General Sort Program

TYPE: Utility

USE: Alternate procedure for sorting a deck of cards

BACKGROUND: Developed to sort urbanized area tract data. To rearrange urbanized areas from alphabetical to FIPS code order.

DESCRIPTION: Sort routine reads blocks to disk, saves a key, sorts the key, and reads from disk according to the key.

INPUT: File to be sorted

OUTPUT: Sorted file

STORAGE: IDA Card Deck S-2

DOCUMENTATION: Attached sheets; see Run S-24 for example of use

LANGUAGE/SYSTEM: RUN(FORTRAN)/6400 SCOPE

COMMENTS: Direct, simple FORTRAN code, but requires random access storage capacity

Preceding Page BLANK - NOT FILMed

This routine sorts blocks of records according to a key. A group of records, here 3000, is read into core storage. Either the whole group is written onto disk, or if the key changes, each key group is written separately. The keys are sorted and records are read from disk to core by the key and then written on the sorted file. This routine is only efficient if the total number of groups is small compared to the total number of records.

To sort different types of records some program statements must be changed. These are flagged in the listing. One position in the array IAR must contain the key and only the key. This position is set by the format stacement in the subroutines REED and RITE.

```
      PROGRAM SNORT( TIN=4002, TOUT=4002, TAPE2=1,OUTPUT)
C
C     SET FOR FILE TO BE SORTED WITH EACH RUN*********************
      COMMON    IAR(13,3001),KEYST(500),KEYNO(500),KEYOR(500),IND(500)
      COMMON/SEND/MT,MU
C
      MU = 4LTOUT
      MT = 3LTIN
      MQ = 6LOUTPUT
C     SET FOR FILE TO BE SORTED WITH EACH RUN*********************
      KEYO = 5H
      MAXKEY = 500
      MAXDIM = 3000
      IW = 13
      IK = 13
      ILIST = 0
      ILIST = 1
      MAXCNT = 2000
      MAXCNT = 9999999

      CALL OPENMS(2,IND,501,0)
      ICNT = 0
      IWCNT = 0
      IIN = 0
      IST = 0
      I = 0
      K = 1
      WRITE(MQ,5)
5     FORMAT(1H1, /////,10X,*RUN OF GROUP SORT PROGRAM *,///,
     1 *   LISTED BELOW ARE GROUP NO-FINAL RECORD IN GROUP NO -SORT KEY
     2- NO RECORDS IN GROUP*,//)
C
10    CONTINUE
      I = I + 1
      IF(I .LE. MAXDIM) GO TO 11
      IBIG = 1
      GO TO 30
11    CONTINUE
      CALL REED(IAR(1,I),IW)
C
      IF(EOF,MT) 22,23
22    CONTINUE
      IIN = 1
      GO TO 30
23    CONTINUE
      ICNT = ICNT + 1
      IF(ICNT .LT. MAXCNT) GO TO 29
      IIN = 1
      GO TO  30
29    CONTINUE
      IF(KEYO .EQ. IAR(IK,I)) GO TO 10
      IF(IST .NE.0) GO TO 25
      KEYO = IAR(IK,I)
      IST = 1
      GO TO 10
25    CONTINUE
      IBIG = 0
C
30    CONTINUE
      IWN = IW*I
      CALL WRITMS  (2,IAR(1,1),IWD,K)
```

IAR should be as long as possible consistent with core KO? storage requirements

Tentatively

IK locates the word in IAR to set key

Reads New Record

to write a group

write a group to disk

2

```
      KEYST(K) = KEYO
      KEYNO(K) = I - 1
      IF (ILIST .NE. 1) GO TO  39
      WRITE(MO,39) K,ICNT,    KEYST(K), KEYNO(K)
36    FORMAT(1H , 20X, I6,5X,I10,5X,A10,I10)
39    CONTINUE
      IF(IIN .EQ.1) GO TO 50
      K = K + 1
      IF(K  .LE. MAXKEY) GO TO  34
      WRITE(MO,35) ICNT
35    FORMAT(///,1H0, * ERROR STOP AFTER READING *,I6, *RECORDS SINCE TO
     1O MANY GROUPS FORMED -------------*  )
      STOP 34
34    CONTINUE
      IF (IBIG .EQ.1) GO TO 32
      KEYO = IAR(IK,I)
      DO 31 J = 1,IW
      IAR(J,1) = IAR(J,I)
31    CONTINUE
      I = 1
      GO TO 10
32    CONTINUE
      I = 0
      GO TO 10
C
C
50    CONTINUE
      CALL ORDNS(KEYST,KEYOR,K)
      KK = 0
C
60    CONTINUE
      KK = KK + 1
      IF(KK .GT. K) GO TO 100
      L = KEYOR(KK)
      IWD = IW*KEYNO(L)
      CALL READMS(2,IAR(1,1), IWD,L)
      LL = 0
C
70    CONTINUE
      LL = LL + 1
      IF(LL .GT. KEYNO(L))GO TO 60
      CALL RITE(IAR(1,LL),IW)
      IWCNT = IWCNT + 1
      GO TO 70
100   CONTINUE
C
      ENDFILE MU
      ENDFILE MU
      WRITE(MO,101) ICNT,IWCNT
101   FORMAT(///,1H0,2X ,*TOTAL RECORDS READ =*,I10,*   TOTAL RECORDS WR
     1ITTEN =*,I10,///)
      STOP 6400
      END
      SUBROUTINE REED(IAR,N)
      COMMON/SEND/MT,MU
      DIMENSION IAR(N)
      READ(MT,21) IAR
C     SET FOR FILE TO BE SORTED WITH EACH RUN********************
21    FORMAT(12A10,A5)
      RETURN
      ENTRY RITE
```

All read now order keys

Read from storage and write to output fil.

Input fil

Output fil.

```
WRITE(MU,21) IAR
RETURN
END
```

COMPUTER PROGRAM DESCRIPTION

NAME: BOX

SYNOPSIS: Produce GBBA, GBBC, GBBD files

TYPE: Single Use

USE: To put Medlist Block Group Records into Standard File format

BACKGROUND: Part of Standard File conversion. This program developed instead of use of MESHFL to save computer time by producing several output files at once.

DESCRIPTION: Direct conversion of file format. Maximum and minimum latitudes for each urbanized area found. Corrections added.

INPUT: Medlist files

OUTPUT: Standard Format Files GBBA, GBBC, GBBD

STORAGE: IDA Card Deck S-3

DOCUMENTATION: Attached sheets; see run S-27

LANGUAGE/SYSTEM: RUN(FORTRAN)/6400 SCOPE

COMMENTS: File GBBD contains only header type data; Data are read into K buffer and written from M buffer to allow attaching TYPE of next record, and for corrections.

PrecedING Page BLaNK - NOT FILMed

```
      PROGRAM BOX(INPLT,OUTPUT,TIN, TOUTA,TOUTB,TOUTC,TOUTD)
C
C     TO PRODUCE MEDLIST GBBA,GBBB,GBBC,AND GBBD RECORDS FIND MIN AND
C     MAX LATTITUDES AND ADU CORRECTIONS TO DATA
C
C
      DIMENSION ISICS(150),  IUAC(150), ITRTC(150), IPOPH(150), FLATO(15
     10) , FLONO(150),  FLATH(150), FLONH(150), ISTCC(150)
      DIMENSION IWVC(20),IWVU(20)
      MP = 5LINPUT
      MQ = 6LOUTPUT
      MT = 3LTIN
      MA = 5LTOUTA
      MB = 5LTOUTB
      MC = 5LTOUTC
      MD = 5LTOUTD
      IEND = 0
      IWVC(1) = 5H54009
      IWVC(2) = 5H54029
      IWVC(3) = 5H54011
      IWVC(4) = 5H54099
      IWVC(5) = 5H54039
      IWVC(6) = 5H54051
      IWVC(7) = 5H54069
      IWVC(8) = 5H54079
      NWV = 8
      DO 5 I =  1,8
      IWVU(I) = 0
5     CONTINUE
      IWVCO = 5H54009
      MSKA = 77770000000000000000B
      MSKB = 40370000000000000000B
      SX = 0.
      SY = 0.
      SXX = 0.
      SYY = 0.
      SXY = 0.
      FLAMX = 0.
      FLOMX = 0.
      FLAMN = 99999.
      FLOMN = 99999.
      IPOPT = 0
      IPOPR = 0
      NRCUA = 0
      ITPSH = 4HGBBA
      ITPSHF= 4HGBBA
      IFORCT = 0
      JTRFCT = 0
      ITPTR = 4HGBBC
      ITPTRF = 4HGBBC
      JSTCT = 0
      ISTPT = 0
      IUACO = 4H
      JUACO = 4H
      KHDT = 4HGBBD
      KHDTN = 4HGBBD
      KTPCT = 0
      KTPCR = 4HGBBB
      KTPCN = 4HGBBB
      KHDCT = 0
      NREC = 4
```

Initialize

```
      KNT = 0
      KNTMX = 1000
      KNTMX = 2000
      KNTMX = 2700
      KNTMX = 999999
      KNTMN = 1400
      KNTMN = 0
C
C
      WRITE(MO,17)
17    FORMAT(1H1 ,////////,10X,* LIST OF CORRECTIONS TO INPUT TAPE G8BB I
     1N THIS RUN OF BOX PROGRAM*,/////)
```

Put correction in core

```
      I = 0
      J = 0
10    CONTINUE
      I = I + 1
      READ(MP,12) ISTCC(I),IUAC(I), ITRTC(I), IPOPH(I), FLONO(I),
     1FLATO(I), FLUNH(I), FLATH(I)
12    FORMAT(A5,8X, A4,12X,A6,7X,I6,6X,F7.4,F6.4,    F7.4,F6.4)
      WRITE(MO,11)ISTCC(I),IUAC(I), ITRTC(I), IPOPH(I), FLONO(I),
     1FLATO(I), FLUNH(I), FLATH(I)
11    FORMAT(1H , A5,8X,A4,12X,A6,7X,I6,6X,2F8.4,1X,2F8.4)
      IF(ISTCC(I) .NE. 5HSTOPP) GO TO 13
      ILNG = I - 1
      IJ = J
      WRITE(MO,18)
18    FORMAT(1H1,/////,10X,* NOW BEGIN PROCESSING INPUT TAPE*,////////)
      GO TO 20
13    CONTINUE
      IF(J .GT.0) GO TO 14
      ISTCS(1) = ISTCC(I)
      J = 1
      GO TO 15
14    CONTINUE
      DO 16 JJ = 1,J
      IF(ISTCC(I) .EQ. ISTCS(JJ)) GO TO 15
16    CONTINUE
      J = J + 1
      ISTCS(J) = ISTCC(I)
15    CONTINUE
      GO TO 10
C
C
```

Read a New Record

```
20    CONTINUE
      READ(MT, 21) KFRNX,KNFCUR,KNFNXT,KRCTNX,KNBFPA,KNBFPB,
     1  KDA,KDB,KDC,KDD,KDE,
     2 KTRCT,KBLGP,KED,KHSE,KPOP, XLAT,XLON ,KCENT
21    FORMAT(A1,A4,1X,A4,      A5,1X,A5,A4,1X,4A10,A7,A6,1X,A1,1X,A5,
     1 2I8,2F8.4,1X,A5)
      IF(EOF,MT) 22,23
22    CONTINUE
      IEND = 1
      ITPSHF = 4HSTOP
      ITPTRF = 4HSTOP
      KHDTN = 4HSTOP
      KTPCN = 4HSTOP
      GO TO 24
```

West Virginia agony

```
23    CONTINUE
      ITM = KNBFPA .AND. MSKA
      IF(ITM .NE. MSKA) GO TO 120
      IF(KNBFPA .EQ. 5H54039) GO TO 130
```

[illegible] FIPS [illegible] record [illegible] area or [illegible]

```
      IF(KNBFPA .EQ. IWVCO) GO TO 125
      DO 126 I = 1,H
      IF(IWVCO .EQ. IWVC(I)) GO TO 127                West Virginia records
126   CONTINUF                                         appear twice, Only save
      WRITE(MO,123, KNBFPA,KNBFPB, KTRCT,IWVCO
128   FORMAT(1H0 ,*///-IN WVA NO OLD COUE MATCH AT*,A5,A4,1X,A6,A5 )  one of these
      IWVCO = KNBFPA
      GO TO 125
127   CONTINUF
      IWVU(I) = 1
      IWVCO = KNBFPA
      GO TO 125
130   CONTINUF
      IF(KTRCT .NE. 6H0107   .OR. KED   .NE. 5H0055  ) GO TO 125
      IWVU(5) = 1
      GO TO 125
125   CONTINUE
      DO 121 I = 1,H
      IF(KNBFPA .EQ. IWVC(I)) GO TO 122
121   CONTINUF
      WRITE(MO,123)  KRCTNX,KNBFPA,KNBFPB,KTRCT,KBLGP,KED
123   FORMAT(1H0, *///IN WVA RECORD DOESNT MATCH AT*,A5,A5,A4,1X,A6,A1,
     1A5)
      GO TO 120
122   CONTINUF
      IF(IWVU(I) .NE. 1) GO TO 61
      WRITE(MO,62) KNBFPA,KNBFPB,KRCTNX,KTRCT,KBLGP,KED,I,IWVU(I)
62    FORMAT(1H ,* W-VA. RECORD DROPPED AT *, A5,A4,2X,A5,2X,A6,A1,
     1 A5,* FLAG =*, I3,I3)
      GO TO 20
61    CONTINUF
      WRITE(MO,63) KNBFPA,KNBFPB,KRCTNX,KTRCT,KBLGP,KED,I,IWVU(I)
63    FORMAT(1H ,* W-VA. RECORD --KEPT- AT *, A5,A4,2X,A5,2X,A6,A1,
     1 A5,* FLAG =*, I3,I3)                                West Virginia agony
120   CONTINUF
      IF(KPOP .EQ. 0) GO TO 20  Delete 0 popn tracts
24    CONTINUF
      KNT = KNT + 1                                        End procedure
      IF(KNT .LT. KNTMN) GO TO 20
      IF(KNT .LE. KNTMX) GO TO 25
      IEND = 1
      ITPSHF = 4HSTOP
      ITPTRF = 4HSTOP
      KHDTN = 4HSTOP
      KTPCN = 4HSTOP
25    CONTINUF
      IF (ISTRT .EQ.1 ) GO TO 28
      ISTRT = 1
      IUACO = KNBFPB
      JUACO = KNBFPB
      MNBFPX = KNBFPA
      MNBFPY = KNBFPB
      GO TO 51
C
28    CONTINUF                                             Look for corrections
      DO 32 I = 1,IJ
      IF(MNBFPA .EQ. ISTCS(I)) GO TO 34
32    CONTINUF
      GO TO 33
34    CONTINUF
C     POSSIBLF MATCH, NOW TRY TRACT,ETC.
```

```
      DO 35 I = 1,ILNG
      IF(MNBFPA, NE. ISTCC(I)) GO TO 35
      IF(MTRCT .NE. ITRTC(I) ) GO TO 35
      ITM = KTPCT + 1
      WRITE(MO,36) IPOP,IPOPH(I), ISTCC(I), IUAC(I),ITRTC(I), FLAT,FLON,
     1 FLATH(I),FLONH(I)   ,KNT,ITM
36    FORMAT(1H0,*----- RECORD CORRECTION/OLD-NEW POP =*,2I8,*  STCTY-UA
     1C-TRACT CODE =*, A5,2X,A4,2X,A6, ,/,*     OLD/NEW(LAT/LON) = *,
     2  2F9.4,5X,2F9.4, *   ALL AT IN/OUT RECORD COUNT = *,2I9)
      IF(IPOP .NE. IPOPH(I)) GO TO 35
      FLON = FLONH(I)
      FLAT = FLATH(I)
      GO TO 37
35    CONTINUF
37    CONTINUE
33    CONTINUF
      IF(ICENT .NE.5H36047) GO TO 65
      IF(FLAT .LT.42.) GO TO 65
      WRITE(MO,66)
66    FORMAT(1H0, *  REPLACE LARGE NEW YORK LATITUDE----//////////*)
67    FORMAT(1H  ,* ---REPLACE FROM--*, A5,A4,2X,A6,A1,A5,2X,A5,
     1 2X,I9,2X,2F8.4)
      WRITE(MO,67) MNBFPA,MNBFPB, MTRCT, MBLGP,MED,ICENT,IPOP,
     1 FLAT,FLON
      FLAT = 40.7504
65    CONTINUF
C
      KTPCT = KTPCT + 1
      WRITE(MD,21)  MFBNX,KTPCR,KTPCN, KTPCT, MNBFPA,MNBFPB,
     1 MDA,MDB,MDC,MDD,MDE ,
     2 MTRCT,MBLGP,MED,IHSE,IPOP, FLAT,FLON ,ICENT
C
      JSTCT = JSTCT + 1
      WRITE(MA,31) MFBNX,ITPSH,ITPSHF, JSTCT,MNBFPA,MNBFPB,
     1 IPOP,IHSE,FLAT,FLON, MTRCT,MBLGP,MED,ICENT
31    FORMAT(A1,A4,1X,A4,I5,1X,A5,A4,  1X,I8,I7,2F8.4,1X,A6,1X,A1,1X,
     1  A5,1X,A5)
C
      IF(IEND .EQ.1) GO TO 47
      IF(KNBFPB .EQ. JUACO) GO TO  45
47    CONTINUE
      IF(IFORCT .EQ.3) GO TO 44
      NREC = 3
      IPOPD = 0
      FLATD = 0
      FLOND = 0
      IPOPC = IPOP
      FLATC = FLAT
      FLONC = FLON
      IF(IFORCT .EQ.2) GO TO  46
      NREC = 2
      IPOPC = 0
      FLATC = 0
      FLONC = 0
      IPOPB = IPOP
      FLATB = FLAT
      FLONB = FLON
      IF(IFORCT .EQ.1) GO TO 46
      NREC = 1
      IPOPB = 0
      FLATB = 0
```

now don't output file

5

```
      FLONB = 0
      IPOPA = IPOP
      FLATA = FLAT
      FLONA = FLON
      GO TO 46
45    CONTINUE
      IFOPCT = IFORCT + 1
      GO TO  (41,42,43,44),IFORCT
41    CONTINUE
      IPOPA = IPOP
      FLATA = FLAT
      FLONA = FLON
      GO TO 48
42    CONTINUE
      IPOPB = IPOP
      FLATB = FLAT
      FLONB = FLON
      GO TO 48
43    CONTINUE
      IPOPC = IPOP
      FLATC = FLAT
      FLONC = FLON
      GO TO 48
44    CONTINUE
      IPOPD = IPOP
      FLATD = FLAT
      FLOND = FLON
46    CONTINUE
      IFOPCT = 0
      JTRECT = JTRECT + 1
      WRITE(MB, 49) MFBNX,ITPTR,ITPTRF,JTRECT, MNBFPA,MNBFPB,
     1NREC, IPOPA, FLATA,FLONA,IPOPB,FLATB,FLONB,IPOPC,FLATC,FLONC,
     2 IPOPD,FLATD,FLOND ,ICENT
49    FORMAT(A1,A4,1X,A4,I5,1X,A5,A4,1X, I1,4(I8,F8.4,F8.4) ,1X,A5)
      NREC = 4
      JUACO = KNBFPB
48    CONTINUE
C
C
      IF(IEND .EQ.1) GO TO 84
      IF(MNBFPB .EQ. IUACO ) GO TO  81
84    CONTINUE
      POP = IPOPB
      CGLA = SY/POP
      CGLO = SX/POP
      CY = 69.17352282
      CX =-69.17352282 *COS(CGLA*3.14159265/180.)
      SXX = SXX/POP
      SXY = SXY/POP
      SYY = SYY/POP
      XX = SXX - CGLO*CGLO
      YY = SYY - CGLA*CGLA
      XY = SXY - CGLA*CGLO
      XX = XX*CX*CX
      YY = YY*CY*CY
      XY = XY*CX*CY
      SGXX = SQRT(ABS( XX))
      SGYY = SQRT(ABS( YY))
      SGXY = SQRT(ABS( XY))
      IF(  XY .LT. 0.) SGXY = - SGXY
      IF (ABS(YY - XX) . GT . 0.00001) GO TO 52
```

Summary statistics
for [illegible]

```
         TTHETA = .5*3.14159265
         GO TO 53
52       CONTINUE
           TTHETA = ATAN(2.*XY/(YY - XX))
53       CONTINUE
         THETA = TTHETA/2.
         SINO = SIN(THETA)
         COSO = COS(THETA)
         SINS = SINO*SINO
         COSS = COSO*COSO
         VARXX = XX*COSS - 2.*SINO*COSO*XY + YY*SINS
         VARYY = YY*COSS + 2.*SINO*COSO *XY + XX*SINS
         IF( VARXX .LT. VARYY) GO TO 55
         VARBB = VARXX
         VARLL = VARYY
         ALPHA = .5*3.14159265    + THETA
         GO TO 58
55       CONTINUE
         VARLL = VARXX
         VARBB = VARYY
         IF( THETA .LT. 0.) GO TO 56
         ALPHA = THETA
         GO TO 58
56       CONTINUE
         ALPHA = 3.14159265 + THETA
58       CONTINUE
         ALPHAD = ALPHA*180./3.14159265
         THN = 0.5*3.14159265 - ALPHA
         SGBB = SQRT(ABS(VARBB))
         SGLL = SQRT(ABS(VARLL))
         IF( VARLL .LT. 0.) SGLL = 0.
         KHDCT = KHDCT + 1
         WRITE(MO,83) MFANX, KHDT, KHDTN, KHDCT,MNBFPX,MNBFPY, IPOPT,
        1 CGLA,CGLO, FLAMN,FLAMX,FLOMN,FLOMX,NRCUA   ,ICENT
        2 ,SGBB,SGLL,ALPHAD
83       FORMAT(A1,A4,1X,A4,I5,1X,A5,A4,1X, I9,2F8.4,1X,2F8.4,1X,2F8.4,I5,
        1 1X,A5, 1X,2F7.4,F7.2)
         WRITE(MO,92) MNBFPX,MNBFPY, KHDCT, JSTCT,JTRECT, IPOPT, IPOPR,
        1 CGLA,CGLO, FLAMN,FLAMX, FLOMN,FLOMX, NRCUA,ICENT, JRAN,KRAN,JRON,
        2 KRON, JRAX,KRAX,JROX,KROX, JBAN,JBON,JBAX,JBOX
92       FORMAT( 1H0, A5,A4, 3I7, 2I9, 2F9.4, 2X,2F9.4,2X,2F9.4,
        1 I6,2X,A5,/,8I8,5X, 4(A6,1X))
         SX = 0.
         SY = 0.
         SXX = 0.
         SXY = 0.
         SYY = 0.
         FLAMX = 0.
         FLOMX = 0.
         FLAMN = 99999.
         FLOMN = 99999.
         IPOPT = 0
         IPOPR = 0
         NRCUA = 0
         IUACO = MNBFPB
         MNBFPX = MNBFPA
         MNBFPY = MNBFPB
81       CONTINUE
         IPOPT = IPOPT + IPOP
         IF(FLAT .LT. 0.001 .OR. FLON .LT. 0.001) GO TO 91
         IPOPR = IPOPR + IPOP
```

Output statistics

Initialize for new O.A.

Collect Statistics

```
         POP = IPOP
         SX = SX + FLON*POP
         SY = SY + FLAT*POP
         SXX = SXX + FLON*FLON*POP
         SYY = SYY + FLAT*FLAT*POP
         SXY = SXY + FLON*FLAT*POP
         IF(FLAT.GE. FLAMN) GO TO 101
         FLAMN = FLAT
         JRAN = JSTCT
         KRAN = JTRECT
         JBAN = MTRCT
101      CONTINUF
         IF(FLON .GE. FLOMN) GO TO 102
         FLOMN = FLON
         JRON = JSTCT
         KRON = JTRECT
         JBON = MTRCT
102      CONTINUF
         IF(FLAT .LE. FLAMX) GO TO 103
         FLAMX = FLAT
         JRAX = JSTCT
         KRAX = JTRECT
         JBAX = MTRCT
103      CONTINUF
         IF(FLON .LE. FLOMX) GO TO 104
         FLOMX = FLON
         JROX = JSTCT
         KROX = JTRECT
         JBOX = MTRCT
104      CONTINUF
91       CONTINUF
         NRCUA = NRCUA + 1
         IF(IEND .EQ.1) GO TO 150
C
51       CONTINUE
         MFBNX = KFBNX
         MNFCUR = KNFCUR
         MNFNXT = KNFNXT
         MRCTNX = KRCTNX
         MNBFPA = KNBFPA
         MNBFPB = KNBFPB
         MTRCT = KTRCT
         MBLGP = KBLGP
         MED = KED
         IHSE = KHSE
         IPOP = KPOP
         FLAT = XLAT
         FLON = XLON
         ICENT = KCENT
         MDA = KDA
         MDB = KDB
         MDC = KDC
         MDD = KDD
         MDE = KDE
         GO TO 20
C
C
150      CONTINUF
         ENDFILE MA
         ENDFILE MA
         ENDFILE MB
```

Advance Buffers

```
      ENDFILE MB
      ENDFILE MC
      ENDFILE MC
      ENDFILE MD
      ENDFILE MD
      WRITE(MO,151)
151   FORMAT(////,20X,*++++++++++++END OF RUN++++++++++++*)
      WRITE(MO,152)
152   FORMAT(1H1)
      STOP 6400
      END
```

Correction Data

```
51059020    78840REMAINDER OF4015    0055 0    606   771458377861 771458387861
51059020    78840REMAINDER OF4020    0056 0   1028   771569377706 771569387706
51059020    78840REMAINDER OF4023    0057 0    304   771695377489 771695387489
51059020    7    REMAINDER OF4023    0057 1   1935   771695377489 771695387489
51059020    78840REMAINDER OF4023    0057B1     11   771695377489 771695387489
51059020    7    REMAINDER OF4023    0057B0    467   771695377489 771695387489
51059020    7    REMAINDER OF4028    0061 1   2816   772458376992 772458386992
51161005    76800REMAINDER OF0312    0058B0    646   788328384920 798328372920
51760005103536760RICHMOND      100901 0203B0    341   765933375855 775933375855
51760005103536760RICHMOND      100901 0204B0     24   765972375386 775972375386
34027135    75601REMAINDER OF046102 0755 0   2217   746997408304 746907408304
34027135    75601REMAINDER OF046102 0755B0   1306   746997408304 746907408304
34033042315559160PENNSVILLE 20216    0122B0    183   755505400710 755505396710
36081005250535601NEW YORK CIT0091  1        0   1057   739310447680 739310407680
36081005250535601NEW YORK CIT0091  9        0      2   739310447680 739310407680
36081005250535601NEW YORK CIT0099  1        0     14   739230447750 739230407750
15003010011033320HONOLULU      0114   0540 0     31  17200002550001579100213000
08041015024031720COLORADO SPR0017   0125 0    170  10581313884861048131388486
08101025108536560PUEBLO        0005   0156 0     59  14611108294301046111382943
04013040035356200SUN CITY >U+0717  1        0   2327  11293203359181122932335918
04013057030546200SCOTTSDALE    2175  9        0    125  11293403348751119340334875
04013058036046200TEMPE         3194   0264B0     23  11290223338801119022333880
04013058036046200TEMPE         3194   0264C0    106  11289053339031118905333903
04013058036046200TEMPE         3199   0265B0      6  11291893337271119189333727
04013058036046200TEMPE         3199   0265C0      5  11293283337211119328333721
04013058    7    REMAINDER OF3194    0264 1     12  11289603338181118960333818
04013058    7    REMAINDER OF3199    0265 1    289  11290363336881119036333688
04013058    7    REMAINDER OF3199    0266 1     30  11294003336991119400333609
48029035374537240SAN ANTONIO 1521    0091 0     29   984736292494 984736293494
48061020132033230HARLINGEN     0105   0045 0   1107   947054261964 977054261964
48167015157032919GALVESTON     1250  2        0   1592   955749295020 948549295020
48439010150032800FORT WORTH    0139   0070 0      0   973059329368 973059328368
48439015305042800NORTH RICHLA013202 0043 0      0   971905329740 971905328740
42049125    72360MILLCREEK TW0106    0190 0    891   801507420969 801507420969
42077010016530240ALLENTOWN     0021  3        0   1381   754866408186 754866406186
42077010016530240ALLENTOWN     0021  4        0    890   754889408187 754889406187
42077010016530240ALLENTOWN     0021  5        0    761   754924408185 754924406185
42077010016530240ALLENTOWN     0023  6        0   1320   755088408119 755088406119
42077010016530240ALLENTOWN     0023  7        0   1287   755036408134 755036406134
42077010016530240ALLENTOWN     0023  8        0    912   754985408155 754985406155
42133265    70280SPRINGETTSBU0101021        0     10   771626400039 766626400039
42133265    79280SPRINGETTSBU0101022        0    676   771239399964 767239399964
19061040121532201DUBUQUE       000299 9900 0     23   909571424274 906571425274
01089025093533440HUNTSVILLE    0010   0099 0    445   863747340454 865747347454
20091036190543760MISSION HILL0509  3*       0      1   916185390080 946185390080
20091036190543760MISSION HILL0509  3**      0      5   916185390080 946185390080
25009035    7    GEORGETOWN T2651    2572 1    703   708318427792 709318427792
25009035    74160GEORGETOWN T2651    2572 0    260   708318427792 709318427792
25009095    7    METHUEN TOWN2521    2554 1   1168   712307427000 712207427000
25009095    74160METHUEN TOWN2521    2554 0    578 I-244 712307427000 712207427000
```

```
1703114515054160IFULTON        8263  1*    0   181      8858194l0185 8758194161185  7
1703114515054160IFULTON        8263  1*    0   209      885819416185 875819416185
170311451505416010ULTON        8264  1     0  1357      885804410338 875804416338
170311451505416010ULTON        8264  2     0  1445      885855410281 875855416281
170311451505416010ULTON        8264  3     0  2834      885877410336 875877416336
17147120     7788UREMAINDER OF0001  9*    0   629      896527394957 896527398325
17197010552041601SIEGER        8838  3     0    38      876286444102 876286414402
180590200580434800CUMBERLAND  4102  1     0   224      869519397810 859519397810
180590400580434800CUMBERLAND  4108  1     0   255      869537397781 859537397781
06013020     7    REMAINDER OF3031    0022 1   789      122629538003212162753800032
060711152005144800NTARIO       0021    0364 0     9      11756753407311175875340731
060711152005144800NTARIO       0022    0365 0     0      117556934066311758693406663
06071150245037279SAN BERNARDI0045   8*    0     6      117406634234411740663413444
0607311024753732OSAN DIEGO    0085101       0  1898      117157232485111171572325851
0607311024753732OSAN DIEGO    0085102       0  1874      117160132498111171601325981
0607311024753732OSAN DIEGO    0085103       0  1579      117165432486811171654325868
0607311024753732OSAN DIEGO    0085109       0    34      .117160932490011171609325900
061110953004360000VENTURA >SAN0012    0092 0   684      119310434279811913043427 98
21047010116034280LEXINGTON    0003    0039 0  1230      843566380189 844506380189
21047010116034280LEXINGTON    0018    0083 0   906      846098380213 845098380213
49035045087037160SALT LAKE CI0002  1       0   366      111953341279011195334077 90
49035045087037160SALT LAKE CI0003011       0  1638      111953341279011195334077 90
32003020006534120LAS VEGAS    000102 0096 0  1061      115625336251711522533615 17
32031015009036720RENO         0018    0070 0  1338      119941039650911984103955 09
32031020010046720SPARKS       0019    0032 0   909      119756539300711975653953 07
33011110177035350NASHUA       950108 0283 0   331      714667427048 714667427408
33015160     74160SALEM TOWN  0003    0144 0  2259      712335427748 712235427748
24005075     70720REMAINDER OF451802 0174 0   713      763813393160 763913393160
24005075     70720REMAINDER OF451802 017480  1630      763813393160 763913393160
24027010032550720ELLICOTT CIT6023  9       0  1677      768607392436 768507392436
24005040     7    REMAINDER OF4089  9*    1   523      766335394853 766335394753
24005040     70720REMAINDER OF4089  9*    0   263      766335394853 766335394753
24033070     70840REMAINDER OF8004049      0   621      767571389851 767751389851
13021020172534680MACON        0127    0131 0   633      836467331035 836467328035
13089010     70520REMAINDER OF0229  1     0   383      842727357760 842727337760
13089020     70520REMAINDER OF0212042      0    79      843066358777 843066338777
13121025026040520EAST POINT   0112028      0   869      844285340543 844285336543
13245005016530600AUGUSTA      0012    0087 0   927      816801334499 819801334499
13215005066031800COLUMBUS     0012    0059 0   959      849400328172 849400325172
13215005066031800COLUMBUS     0030    0135 0   755      849226328626 849226324626
13215005     71800REMAINDER OF0004    0141 0   199      849899328013 849899325013
13295005     71560REMAINDER OF0201    0008 0   953      852730348409 852730349409
22073025143045200WEST MONROE 0051    0006 0   813      921429326968 921429324968
22073025143045200WEST MONROE 0052    0007 0   619      921500327031 921500325031
55025030     74720BLOOMING GR0026029*      0     4      892919431892 892919431092
55025055     74720BURKE TOWN  002602 0447G0     5      893000432500 893000431500
55025055     7    BURKE TOWN  002602 0447G1   312      893000432500 893000431500
55025055     7    BURKE TOWN  0026029*     1     6      892919431892 892919431092
55101080283044600WIND POINT   0015  1*    0  1251      880069431255 877069428255
12011010042042680COOPER CITY 0704  9     0  1973      802686265639 802686260639
12011010042042680COOPER CITY 0704  9     0   562      802686265639 802686260639
12019020     7    REMAINDER OF9501    180181   238      817601261480 817601301480
12019020     73600REMAINDER OF9501    180180   556      817601261480 817601301480
12103025053047060DUNEDIN      0269011*     0    22      827526290442 827546280442
12103025053047060DUNEDIN      0269011**    0   172      827526290442 827526280442
12103025053047060DUNEDIN      0269019*     0     6      827535290421 827535280421
12103025053047060DUNEDIN      0269019**    0   272      827535290421 827535280421
12103025053047060DUNEDIN      0271021*     0  1874      827722290420 827722280420
12103025053047060DUNEDIN      0271021**    0    18      827722290420 827722280420
12103025053047060DUNEDIN      027103 0015 0   194      828119290451 828119280451
12103025053047060DUNEDIN      0272    0014 0   336 I-245 827826290533 827826280533
```

```
12103025053047060DUNEDIN      0272  4*    0      4          827660290723 827660280723  70
12103025053047060DUNEDIN      0272  4**   0    219          827660290723 827660280723
12103025     77060REMAINDER OF0269019*    0     17          827535290421 827535280421
2601701501653080BAY CITY      0010  7     0    884          839388434035 839388436035
2616303506803216DETROIT       0007  1     0    355          838082423245 830882423245
39013005     7    CULERAIN TWP0103    0913 1    750          808869401358 807869401358
39013005     79000CULERAIN TWP0103  9     0     59          808869401358 807869401358
40037036     78560REMAINDER OF0201  1     0    261          981637345240 961637360240
4010903618153588OKLAHOMA CIT106903 0844 0    324          956575354988 976575354988
34013         5601NEW YORK    0147               377          841798407789 741798407789
26077         3720KALAMAZOO   0022011              2          845150423549 855150423549
42101         6160PHILADELPHIA018399 2165          0          751288409897 751288399897
36081         5601NEWYORK     0695                66          738675447290 738675407290
36091         5601NEWYORK     0564             16520          737325447210 737345407210
36091         5601NEWYORK     0178                38          738095446772 738095406772
12103         7060 ST PETE    026901               2          827665285328 827665280328
STOPP
```

COMPUTER PROGRAM DESCRIPTION

NAME: MOVEUB

SYNOPSIS: Standard File XBΔΔ editing

TYPE: Single Use

USE: Rearrange records

BACKGROUND: The XBΔΔ file has nodal data. When several urbanized areas are in a single county they are listed together. In six cases the order is different on this file than in Medlist. This rearranges the order.

DESCRIPTION:

INPUT: File XBΔΔ-1

OUTPUT: File XBΔΔ-2; see Run S-30.

STORAGE: IDA Card Deck S-4

DOCUMENTATION: Attached sheet

LANGUAGE/SYSTEM: RUN(FORTRAN)/6400 SCOPE

COMMENTS: The separate cases are rearranged individually by use of two supplementary storage arrays.

```
      PROGRAM MOVEUB(TIN,TOUT)
C
C     TO PUT SEVERAL UB RECORDS FROM LB  -1 IN ORDER OF MEDLIST
C       ON GHBC-1
C
      DIMENSION IAR(13),IARA(13),IARB(13)
      MT = 3LTIN
      MU = 4LTOUT
10    CONTINUE
      READ(MT,21)  IAR (1), INU , (IAR (I),I=2,12)
21    FORMAT(A10,I5,10A10,A3)
      IF(INU .NE. 503 .AND. INU.NE. 4860.AND. INU .NE.4936     Rearrange 2
     1 .AND. INU .NE. 2294  ) GO TO 31
      DO 41 I = 1,13
      IARA(I) = IAR(I)    Fill IARA
41    CONTINUE
      INUA = INU
      READ(MT,21)  IAR (1), INU , (IAR (I),I=2,12)
      INUST = INU
      INU = INUA
      INUA = INUST
      IARSTA = IAR(1)
      IAR(1) = IARA(1)
      IARA(1) = IARSTA
      WRITE(MU,21) IAR (1), INU , (IAR (I),I=2,12)   Write reverse order
      WRITE(MU,21) IARA(1), INUA, (IARA(I),I=2,12)
      GO TO 10
31    CONTINUE                                         Rearrange 3
      IF(INU .NE. 616) GO TO 32
      DO 42 I = 1,13
      IARA(I) = IAR(I)
42    CONTINUE
      INUA = INU
      READ(MT,21)  IAR (1), INU , (IAR (I),I=2,12)
      DO 43 I = 1,13
      IARB(I) = IAR(I)
43    CONTINUE
      INUB = INU
      READ(MT,21)  IAR (1), INU , (IAR (I),I=2,12)
      INUSB = INUB
      INUST = INU
      INU = INUA
      INUA = INUSB
      INUB = INUST
      IARSTB = IAR(1)
      IAR(1) = IARB(1)
      IARB(1) = IARSTB
      WRITE(MU,21) IAR (1), INU , (IAR (I),I=2,12)
      WRITE(MU,21) IARA(1), INUA, (IARA(I),I=2,12)
      WRITE(MU,21) IARB(1), INUB, (IARB(I),I=2,12)
      GO TO 10
32    CONTINUE                                         Rearrange 3
      IF(INU .NE. 641) GO TO 33
      DO 44 I = 1,13
      IARA(I) = IAR(I)
44    CONTINUE
      INUA = INU
      READ(MT,21)  IAR (1), INU , (IAR (I),I=2,12)
      DO 45 I = 1,13
      IARB(I) = IAR(I)
45    CONTINUE
```

Preceding Page Blank - NOT FILMED

```
      INUB = INU
      READ(MT,21)  IAR (1), INU , (IAR (I),I=2,12)
      INUSB = INUB
      INUST = INU
      INU = INUA
      INUA = INUSB
      INUB = INUST
      IARSTB = IAR(1)
      IAR(1) = IARB(1)
      IARB(1) = IARSTB
      WRITE(MU,21) IAR (1), INU , (IAR (I),I=2,12)
      WRITE(MU,21) IARA(1), INUA, (IARA(I),I=2,12)
      WRITE(MU,21) IARB(1), INUB, (IARB(I),I=2,12)
      GO TO 10
33    CONTINUE
      WRITE(MU,21) IAR (1), INU , (IAR (I),I=2,12)
      IF(INU .EQ. 5774)GO TO 60
      GO TO 10
60    CONTINUE
      ENDFILE MU
      STOP 6400
      END
```

COMPUTER PROGRAM DESCRIPTION

NAME: SHUFFL

SYNOPSIS: Standard File Production

TYPE: Single Use

USE: To correct central county code on lost record for an urganized area on GBBD file. Compare population on Medlist and nodal records.

BACKGROUND: Output to assist in correction file codes.

DESCRIPTION: GBBD record is read and then nodal file. Populations are compared.

INPUT: File XBΔΔ-1; Nodal Data; Central County Data

OUTPUT: File GBBD

STORAGE: IDA Card Deck S-5

DOCUMENTATION: Attached sheet; see Run S-31

LANGUAGE/SYSTEM: RUN(FORTRAN)/6400 SCOPE

COMMENTS: Used to find problem urbanized areas where populations do not agree or sizes drastically different. Also to correct error from program BOX.

/

```
      PROGRAM SHUFFL(OUTPUT,TINA,TINB,TOUT)
      MQ= 6LOUTPUT
      MA = 4LTINA
      MB = 4LTINB
      MT = 4LTOUT
C     A  IS POPN, B = GBBD
      ICNT = 0
      SMI = 69.17352282
      WRITE(MQ,5)
5     FORMAT(1H1,////////,20X,*****COMPARE POPN ON XB  -1 AND GBBD-1****
     1 *,/////)
      ICENTO = 5H01055
      IEND = 0
      ITPN = 4H
C
10    CONTINUE
      READ (MB,83) MFBNX, KHDT, KHDTN, KHDCT,MNBFPX,MNBFPY, IPOPT,
     1 CGLA,CGLO, FLAMN,FLAMX,FLOMN,FLOMX,NRCUA  ,ICENT
     2 ,SIGB,SIGL,ALPH
      IF(KHDCT .EQ.247) IEND = 1
      WRITE(MT,83) MFBNX, KHDT, KHDTN, KHDCT,MNBFPX,MNBFPY, IPOPT,
     1 CGLA,CGLO, FLAMN,FLAMX,FLOMN,FLOMX,NRCUA  ,ICENTO
     2 ,SIGB,SIGL,ALPH
83    FORMAT(A1,A4,1X,A4,I5,1X,A5,A4,1X, I9,2F8.4,1X,2F8.4,1X,2F8.4,I5,
     1 1X,A5,1X,2F7.4,F7.2)
      ICENTO = ICENT
C
      IF(ITPN .EQ.4HUH   ) GO TO 34
30    CONTINUE
        READ(MA,32) ITPN
32    FORMAT(6X,A4)
      IF(EOF,MA) 63,64
64    CONTINUE
      IF(ITPN .NE.4HUH   ) GO TO 30
34    CONTINUE
      READ(MA,31) ITPC,ITPN,IFPA,IFPB,NAMEA,NAMEB,NAMEC,IPOPP,FLATP,
     1 FLONP,SIGB,SIGL,ALPH
31    FORMAT(1X,A4,1X,A4,6X,A5,A4,1X,3A8,I9,2F8.4,2F7.3,F7.2)
      IF(EOF,MA) 63,65
65    CONTINUE
      IF(IFPB .NE. MNBFPY) GO TO 36
      GO TO 38
36    CONTINUE
      WRITE(MQ,37) ICNT,IFPA,IFPB,MNBFPX,MNBFPY
37    FORMAT(*   0H 0H*,I7,2X,A5,2X,A5,2X,A4,2X,A4)
      STOP 234
38    CONTINUE
      POPP = IPOPP
      POPP = POPP/1.05
      IPOPP = POPP + .5
      SIGB = 3.*SIGB
      SIGL = 3.*SIGL
      MAXA = (FLAMX-CGLA)*SMI   +.99999
      MINA = (CGLA - FLAMN)*SMI       +.99999
      MDA = (FLAMX-FLAMN)*SMI + .99999
      COSLA = COS(CGLA*3.14159265/180.)
      MAXO = (FLOMX - CGLO)*COSLA*SMI   +.99999
      MINO = (CGLO - FLOMN)*COSLA*SMI + .99999
      MDO = (FLOMX - FLOMN)*COSLA*SMI +.99999
      ICNT = ICNT + 1
63    CONTINUE
```

Read GBBD file

Read Node file

Get UB record

Get max, min coordinates

Preceding Page Blank - NOT FILMed

Test 2
Comparison

```
      WRITE(MW, 41)ICNT,IFPA,IFPB,NAMEA,NAMEB,IPUPT,IPUPP,FLATP,CGLA,
     1 FLONP,CGLO,SIGH,SIGL,ALPH,MAXA,MINA,MDA,MAXO,MINO,MDO
41    FORMAT( I5'1X'A5'A4'1X'A8'A4'2I9'  2X'4F9.4'2F7.3'F7.2'6(1X'I4))
C
      IF (IEND .EQ.1) GO TO 200
      GO TO 10
200   CONTINUE
      ENDFILE MT
      STUP 6400
      END
```

COMPUTER PROGRAM DESCRIPTION

NAME: CRDATK

SYNOPSIS: Produce Standard File

TYPE: Single Use

USE: To punch card images of a sample attack, file

BACKGROUND: Could have been done in MESHFL. Separate program written so core could contain entire attack, for sorting

DESCRIPTION: Attack cards are read. Central county codes are attached and the file is sorted on the resulting FIPS code.

INPUT: Attack Deck

OUTPUT: File IWAD

STORAGE: IDA Card Deck S-8

DOCUMENTATION: Attached sheet; see Run S-32

LANGUAGE/SYSTEM: RUN(FORTRAN)/6400 SCOPE

COMMENTS:

```
         PROGRAM CHDATK(INPUT,OUTPUT,PUNCH,TIN,TOUT)
         DIMENSION ISTCOA(400),IUACA(400)
         DIMENSION ISCCA(1750),JUAA(1750),NAMAA(1750),NAMBA(1750),NAMCA(175
        10), IOLDCA(1750),    FLATA(1750),FLONA(1750), YLDA(1750),IORD(1750)
         DIMENSION ICOMT(20)
         MP = 5LINPUT
         MQ = 6LOUTPUT
         MS = 5LPUNCH
         MT = 3LTIN
         MA = 4LTOUT
         ITYP = 1
         ITYP = 2
         ITYPO = 4
         ITYPO = 3
         ITYPO = 2
         ITYPO = 1
         IRCT = 0
         ITYP = 4HIWAD
         ITYPN = 4HIWAD
         MSKA =   77777777777777777778
         MSKB=  33000000000000000000B
         MSKC=  77000000000000000000B
         MSKD=  55000000000000000000B
         WRITE(MQ,6)
 6        FORMAT(1H1,/////,10X,* PUNCH CARDS FROM OLD CPR WEAPONS FOR SAMPL
        1 IWAD STANDARD FILE*,////)
         MFB = 1H
         IOC = 0
         ICL = 2H
         IAT = 4H
         IN = 0
 10      CONTINUE
         IN = IN + 1
         READ(MP,11) ISTCO,IUAC
 11      FORMAT(5X    ,A5,1X,A4)
         ISTCOA(IN) = ISTCO
         IUACA(IN) = IUAC
         IF(IUAC .NE.4H9320 ) GO TO 10
 20      CONTINUE
         IRCT = IRCT + 1
         IF(ITYP .EQ.2) GO TO 52
         READ(MP,21) JUA, NAMA,NAMB,NAMC,FLOD,FLOM,FLOS,FLAD,FLAM,FLAS,
        1 YLD,IOLDC
 21      FORMAT(1X,A4,5X,2A10,A4,2X,1X,9X,1X, 7F4.0,I5)
         IF(JUA .NE.4HSTOP ) GO TO 41
         IRCT = IRCT - 1
         GO TO 29
 41      CONTINUE
         GO TO 53
 52      CONTINUE
         READ(MT,51) JUA,NAMA,NAMB,NAMC,  FLOD,FLOM,FLOS,FLAD,FLAM,FLAS,
        1 YLD,IOLDC, ICOD,JCOD,(ICOM(I),I=1,6)
 51      FORMAT(1X,A4,2A10,A4,6X,F10.0, 7F4.0,3X, A4,1X,A1,A1,5A10,A3)
         IF(EOF,MT) 54,55
 54      CONTINUE
         GO TO 29
 55      CONTINUE
 C       INSERT HERE WHAT TO DO IF UA CODE IS 0
 53      CONTINUE
         ITM = JUA .AND. MSKC
         IF(ITM .NE.MSKD) GO TO 23
```

Picked Urbanized Area
Central County Code
association

Reads Weapon

Preceding Page Blank - NOT FILMED

```
          JUA = MSKB .OR.(JUA .AND. MSKA)
23        CONTINUE
          FLON = FLOD + FLOM/60. + FLOS/3600.
          FLAT = FLAD + FLAM/60. + FLAS/3600.
          DO 25 I = 1,IN
          IF(JUA .EQ. IUACA(I ))GO TO 26
25        CONTINUE
          ISCC = 5H99999
          GO TO 28
26        CONTINUE
          ISCC = ISTCOA(I)
28        CONTINUE
          ISCCA(IRCT) = ISCC
          JUAA(IRCT) = JUA
          NAMAA(IRCT) = NAMA
          NAMCA(IRCT) = NAMC
          NAMBA(IRCT) = NAMB
          FLATA(IRCT) = FLAT
          FLONA(IRCT) = FLON
          IOLDCA(IRCT) = IOLDC
          YLDA(IRCT) = YLD
C         IF(IRCT .GT. 65) GO TO 29
          IF( JUA .NE. 4H0000 .OR. FLOM .NE.35. .OR . FLAS .NE. 30.) GO TO
         1 20
29        CONTINUE
          CALL ORUNS(ISCCA,IORD,IRCT)
          DO 35 J = 1,IRCT
          K = IORD(J)
          IF(J .EQ.IRCT) ITYPN =4HSTOP
          IOC = IOC + 1
          WRITE(MS,32)  MFB,ITYP,ITYPN,IOC, ISCCA(K), JUAA(K), NAMAA(K),
         1 NAMBA(K),NAMCA(K), ICL,IAT,IOLDCA(K),FLATA(K),FLONA(K),YLDA(K)
          WRITE(MW,32)  MFB,ITYP,ITYPN,IOC, ISCCA(K), JUAA(K), NAMAA(K),
         1 NAMBA(K),NAMCA(K), ICL,IAT,IOLDCA(K),FLATA(K),FLONA(K),YLDA(K)
32        FORMAT(       A1    ,A4,1X,A4,I5,1X,A5,A4,1X,2A10,A4,A2,A4,I4,
         1  2F8.4,F4.0)
35        CONTINUE
          STOP 6400
          END
```

add central county code

Sort on FIPS code

Output file

COMPUTER PROGRAM DESCRIPTION

NAME: TOGRID

SYNOPSIS: Put Block Group Population on Grids

TYPE: SINGLE USE

USE: To produce standard format files QBAA, QBAB, and QBAC.

BACKGROUND:

DESCRIPTION: Population tract date is read and allocated to the nearest grid point. Three (3) types of output files are written.

INPUT: File GBBD for General Urbanized area description
File GBBC for data

OUTPUT:

STORAGE: IDA Card Deck S-9
See, for example, Runs S-36, S-37

DOCUMENTATION: Attached Sheet

LANGUAGE/SYSTEM: Run (FORTRAN)/6400 Scope

COMMENTS:

File Type	Description
QBAA	Six grid points/record are written each point has grid on (I) value, grid N-S (J) value, and population
QBAB	Up to 14 grid points/record are written grid I, J value for left most point is written first. Then grid population values are written for points moving to the right (east) until a grid point with 0 population is encountered when the record is terminated.
QBCA	A map of grid points in each U. A. is produced. This is not strictly a Standard Format Data file.

```
      PROGRAM TOGRID(TINA,TINB,TOUTA,TOUTB,TOUTC,OUTPUT,INPUT)                1
C
      DIMENSION IPNT(20000)
      DIMENSION IARR(6),JARR(6),NPARR(6),KPOPU(20)
      DIMENSION FLATI(4),FLONI(4),KPOPI(4)
      DIMENSION LNVAL(40),LNPRT(140) ,INUMB(10)
      DIMENSION NAMEA(250),NAMEB(250),IACAR(250)
      DATA LNVAL     / 1H ,1H.,1H1,    1H2,1H3,1H4,1H5, 1H6,1H7,1H8,1H9,
     1 1H0,1HA,1HB,1HC,1HD,1HE,  1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,
     2 1HN,1HO, 1HP,1HQ,1HR,1HS,1HT,1HU,1HV,1HW,1HX,1HY,1HZ,1H+,1H*/
      DATA  INUMB / 1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H0  /
      MA = 4LTINA
      MB = 4LTINB
      MT = 5LTOUTA
      MU = 5LTOUTB
      MV = 5LTOUTC
      MO = 6LOUTPUT
      MP = 5LINPUT
C     MA  IS HEADER ON G8BU, MB IS DATA(4 IN 1) ON G8BC
      MFB = 1H
      MFZO = 1H0
      MFHS = 1H1
      IEND = 0
      ISTRT = 1
      MRCT = 4HQBAA
      MRTH = 4HQBAB
      MRTY = 4HQBBA
      MRTYH = 4HQBBB
      MRTMP  = 4HQBCA
      MRTMPH = 4HQBCB
      IBUGCT  = 0
      IBUGA = 0
      IBUGCT = 9999
      IBUGA = 9999
      IBUGMX = 3
      IBUGMX = 99999
      ISKIP = 0
      JOUTY = 0
      JOUTMP = 0
      JOUTCT = 0
      NSSIX = 14
      SMIC = 69.17352282
      SPGX = 2.
      SPGY = 2.
      SPGX = 1.
      SPGY = 1.
      SPGX = 1.6666666667
      SPGX = 0.4
      SPGY = 0.4
      SPGXO = 24./SMIC
      SPGY = 24./SMIC
      DISTX = 1./SPGX
      DISTY = 1./SPGY
      DISTHX = 0.5*DISTX
      DISTHY = 0.5*DISTY
      WRITE(MO,9 )  SPGY,SPGX
 9    FORMAT(  1H1,///// *          RUN OF GRID INPUT PROGRAM*,////,
     1  *   IN THIS RUN THERE ARE *,F8.4, * GRID POINTS NORTH SOUTH AND
     2*,F8.4,* GRID POINTS EAST WEST IN EACH MILE*,///)
      INN = 0
 11   CONTINUE
```

Initialize

] Grid spacing East-West, points/mile
Grid spacing North-South, points/mile

Preceding Page BLANK - NOT FILMED

```
        INN = INN + 1
        READ(MP,12) IACAR(INN),NAMEA(INN),NAMEB(INN)
 12     FORMAT(11X,A4,5X,A10,A8)
        IF(IACAR(INN) .NE. 4H9320) GO TO 11
        GO TO 30
 C
```

Start new U.A

```
 20     CONTINUE
        IBUGCT = IBUGCT + 1
        IF(IBUGCT .GT. IBUGMX) IEND = 1
        DO 14 I = 1,140
        LNPRT(I) = 1H
 14     CONTINUE
```

Bad Header Data

```
        READ(MA,21) MFBNX,IRGD,IRGDN,ICNT,KNBFPA,KNBFPB, IPOPT
       1 ,FLATCG,FLONCG,FLATMN,FLATMX,FLONMN,FLONMX    ,NUREC,KCENT
       2,SIGH,SIGL,ALPH
 21     FORMAT(A1,A4,1X,A4,    A5,1X,A5,A4,1X,    I9,2F8.4,1X,2F8.4,1X,
       1 2F8.4,I5,1X,A5, 1X,2F7.4,F7.2)
        IF(EOF,MA) 22,23
 22     CONTINUE
        GO TO 200
 23     CONTINUE
        IF(IRGDN .EQ.4HSTOP) IEND = 1
        JNBFPA = KCENT
        JNBFPB = KNBFPB
        DO 25 I = 1,INN
        IF(KNBFPB .EQ. IACAR(I)) GO TO 26
 25     CONTINUE
        NAMEAP = 10HUNLISTED
        NAMEBP = 8H
        GO TO 27
 26     CONTINUE
        NAMEAP = NAMEA(I)
        NAMEBP = NAMEB(I)
 27     CONTINUE
 C      SPECIAL MODS FOR 2 1/2 MINUTE GRID*********************** START
        IFLA = FLATCG*24.
        IFLO = FLONCG*24.
        IFLA = IFLA + 1
        IFLO = IFLO + 1
        FLATCG = IFLA
        FLONCG = IFLO
        FLATCG = FLATCG/24.
        FLONCG = FLONCG/24.
        SPGX = SPGXO/(COS(FLATCG*3.14159265/180.))
        DISTX = 1./SPGX
        DISTHX = .5*DISTX
 C      SPECIAL MODS FOR 2 1/2 MINUTE GRID*********************** END
```

Set Grid
Grid centered on population
CG in header

```
        DY = SMTC
        DX = SMTC*COS(FLATCG*3.14159265/180.)
        NGY = (FLATCG-FLATMN)*DY/DISTY+ DISTHY
        NGX = (FLONMX - FLONCG )*DX/DISTX+ DISTHX
        XLX = NGX*DISTX
        XLY = NGY*DISTY
        ITM = (FLATMX - FLATCG)*DY/DISTY+ DISTHY
        MAXY = NGY + ITM + 1
        ITM =(FLONCG - FLONMN)*DX/DISTX+ DISTHX
        MAXX = NGX + ITM + 1
        ISKIP = 0
        ITM = MAXX*MAXY
        IF(ITM .LT.20000) GO TO 106
        WRITE(MO,107) KNBFPA,KNBFPA,MAXX,MAXY
```

```
107    FORMAT(      1H0,*ARRAY TOO SMALL SO SKIP AT*, A5,A4,2X,2I6   )
       ISKIP = 1
106    CONTINUE
C*******BUGGGGGGG
       IF(IBUGCT .GT. 10) GO TO 318
       WRITE(MO,310)  DX,DY,XLX,XLY, NGX,NGY,MAXX,MAXY
310    FORMAT(1H ,* AT 310*,  4F12.4,4I8)
318    CONTINUE
       DO 24 I = 1,MAXX
       DO 24 J = 1,MAXY
       K = I + MAXX*(J-1)
       IPNT(K) = 0
24     CONTINUE
       NNZ = 0
       GO TO 34
C
30     CONTINUE
       IBUGA = IBUGA + 1
       READ(MB,31) KFRMX,IRGC,IRGCN,KCNT,MNBFPA,MNBFPB, NRCH,
      1 (KPOPI(I),FLATI(I),FLONI(I),I=1,4)       ,ICENT
31     FORMAT(A1,A4,1X,A4,I5,1X,A5,A4,1X,I1,4(I8,F8.4,F8.4),1X,A5)
```

Read Population Tract Data

```
       IF(EOF,MB) 32,33
32     CONTINUE
       GO TO 45
33     CONTINUE
       IF(ISTRT .EQ. 0) GO TO 34
       ISTRT = 0
       IUUCO = MNBFPB
       GO TO 20
34     CONTINUE
       IF(MNBFPB .EQ. IUUCO) GO TO 36
       IUUCO = MNBFPB
       GO TO 45
36     CONTINUE
       IF(ISKIP.EQ.1) GO TO 30
       IF(ICENT .EQ.KCENT) GO TO 37
       IF(IBUGA .GT. 30) GO TO 37
       WRITE(MO,38) ICENT,KCENT, MNBFPA,MNBFPB,KNBFPA,KNBFPB
38     FORMAT(1H0,* TROUBBLE-- NO MATCH OF CENTRAL COUNTIES AT--*,6(2XA5)
      1 )
C      STOP 333
37     CONTINUE
       DO 39 I = 1,NRCH
```

Put in Grid

```
       IX = ((FLONCG- FLONI(I)) *DX + XLX)/DISTX + DISTHX
       IX = IX + 1
       IY = ((FLATI(I) - FLATCG)*DY + XLY)/DISTY + DISTHY
       IY = IY + 1
       IF(IX .LE. MAXX .AND. IY .LE. MAXY) GO TO 41
       WRITE(MO,40) KCNT,MNBFPA,MNBFPB, IX,IY,MAXX,MAXY
40     FORMAT(1H0 *---TROUBLE IN SIZE, AT*,I5,  2X,A5,A4,4I9)
       IF(IX .GT. MAXX) IX = MAXX
       IF(IY.GT.MAXY) IY = MAXY
41     CONTINUE
       IF(IX .GE.1.AND. IY.GE.1) GO TO 101
       WRITE(MO,102) KCNT,MNBFPA,MNBFPB,IX,IY,MAXX,MAXY
102    FORMAT(1H0, * GRID POINT TOO SMALL AT *,I5,2X,A5,A4,4I9)
       IF(IX .LT. 1) IX = 1
       IF(IY .LT. 1) IY = 1
101    CONTINUE
C*****BUGGGGGGGG
       IF(IBUGCT .GT.1) GO TO 44
```

4

```
      WRITE(MO,315)KCNT,MNBFPA,MNBFPB, IX,IY,MAXX,MAXY
315   FORMAT(1H0 *---   AT 315 NOW E. AT*,I5,  2X,A5,A4,4I9)
C*******BUGGGGGG
      WRITE(MO, 312) FLONCG,FLONI(I),FLATI(I),FLATCG, XLX,XLY
312   FORMAT( 1H  ,* AT 312  *, 6F12.4)
44    CONTINUE
      J = IX +(IY - 1)*MAXX
      IF(IPNT(J) .EQ.0) GO TO 42
      IPNT(J) = IPNT(J) +  KPOPI(I)
      GO TO 43
42    CONTINUE
      NNZ = NNZ + 1
      IPNT(J) = KPOPI(I)
43    CONTINUE
39    CONTINUE
      GO TO 30
C
C
C     AT THI S POINT ARRAY S FILLED NOW START OUTPUT
C
C     FIRST HEADERS FOR THE CITY
C
45    CONTINUE
C*****BUGGGGGGGG
      IF(IBUGCT .GT. 1) GO TO 343
      DO 340 J = 1,MAXY
      DO 340 I = 1,MAXX
      K = I + (J-1)*MAXX
      WRITE(MO,342)I,J,K, IPNT(K)
342   FORMAT(1H  ,* ARRAAY PT =*,3I5,5X,I10)
340   CONTINUE
343   CONTINUE
      IF(ISKIP.EQ.1) GO TO 26
      JOUTCT = JOUTCT + 1
      WRITE(MT,46) MFRO, MRTH,MRCT , JOUTCT,JNBFPA,JNBFPB, IPOPT,
     1 FLATCG,FLONCG, NGY,NGX,XLY,XLX,MAXY,MAXX
46    FORMAT(A1 ,A4,1X,A4,I5,1X,A5,A4,1X, I9,2F8.4,1X,2I4,1X,2F8.4,
     1 1X,2I4)
      JOUTY = JOUTY + 1
      WRITE(MU,47) MFRO,MRTYH,MRTY, JOUTY, JNBFPA,JNBFPB, IPOPT ,
     1 FLATCG,FLONCG, NGY,NGX,XLY,XLX,MAXY,MAXX
47    FORMAT(A1 ,A4,1X,A4,I5,1X,A5,A4,1X, I9,2F8.4,1X,2I4,1X,2F7.4,
     1 1X,2I4)
      JOUTMP = JOUTMP + 1
      WRITE( MV,48) MFBS,MRTMPH,MRTMP, JOUTMP, JNBFPA,JNBFPB,
     1 NAMEAP,NAMEBP,
     2 JNBFPB,JNBFPA, IPOPT, FLATCG,FLONCG, NGY,NGX, XLY,XLX,
     3 MAXY,MAXX
48    FORMAT( A1, A4,1X,A4,I5,1X,A5,A4,1X,///, 20X, *POPULATION GRID MAP
     1 FOR THE CITY OF *,A10,A8   .//,          *  MAP FOR URBANIZED AREA
     2 NO. *,A4,* WITH STATE COUNTY CODE *,A5,* , AND POPULATION = *,I9,
     3 /,* LATITUDE/LONGITUDE OF C.G. OF POPULATION = *, F8.4,2X,F8.4,
     4  /,* GRID STEPS SOUTH /WEST OF C.G. = *,2I6, * AT DISTANCES OF *,
     5 2F9.4      ,/, * N/S GRID STEPS = *, I6,*  AND E/W GRID STEPS =*,
     6 I6, ///)
      DO 82 I = 1,MAXX
      II = I/10
      IA = I - II*10
      IF(IA .EQ. 0) GO TO 83
      LNPRT(I) = 1H+
      GO TO 82
```

End Population Pro

Write output headers

```
83      CONTINUE
        IF(II .GT. 10) GO TO 84
        LNPRT(I) = INUMB(II)
        GO TO 82
84      CONTINUE
        II = II - 10
        GO TO 83
82      CONTINUE
        WRITE(MV,85)(LNPRT(I),I = 1,MAXX   )
85      FORMAT( 1H ,3X, 131A1)
C
C
C       NOW START AT TOP LINE AND WORK DOWN
C
C
        NINCT = 0
        NSSCT = 0
        IY = MAXY
55      CONTINUE
C       NEW LINE
        IX = 0
54      CONTINUE
        IX = IX + 1
        IF(IX .LE.MAXX) GO TO 52
        IF(IY .EQ. MAXY) GO TO 72
        JOUTMP = JOUTMP + 1
        ITTM = MAXY - IY
        JPR = ITTM /10
        ITM = ITTM - JPR*10
        IF(ITM .EQ. 0) GO TO 86
        WRITE(MV,88) (LNPRT(I),I = 1,MAXX    )
88      FORMAT(1H  ,2X,1H+, 131A1)
        GO TO 87
86      CONTINUE
        WRITE(MV,73) JPR,(LNPRT(I), I = 1,MAXX)
73      FORMAT(1H ,I3,      131A1)
87      CONTINUE
72      CONTINUE
        IY = IY - 1
        IF(IY .LT. 1) GO TO 53
        GO TO 55
52      CONTINUE
        J = (IY - 1)*MAXX + IX
        IF(IPNT(J) .NE.0) GO TO 74
        K = 1
        GO TO 71
74      CONTINUE
        K = IPNT(J)/1000
C******BUGGGGGGGG
        IF(K .GE. 0) GO TO 632
        WRITE(MO,633) K  ,IX,IY,J,IPNT(J)
633     FORMAT(1H   ,* AT LOI WITH NEG K*,  5I10)
        K = 0
632     CONTINUE
        K = K + 2
        IF(K .LE. 12) GO TO 71
        K = IPNT(J)/5000
        K = K + 11
        IF(K .LE. 38) GO TO 71
        K = 40
71      CONTINUE
```

Write Map

Writ QBAB file

```
      LNPPT(IX) = LNVAL(K)
C
      IF(IPNT(J) .EQ.0) GO TO 54
      IF(NSSCT .NE. 0) GO TO 91
      NSSCT = 1
      KPOPO(NSSCT) = IPNT(J)
      IY0 = IY
      IX0 = IX
      IXL = IX
      GO TO 92
91    CONTINUE
      ITM = IXL + 1
      IF(ITM .EQ.IX) GO TO 93
      JOUTY = JOUTY + 1
      WRITE(MU,81) MFR,MRTY,MRTY,JOUTY,JNBFPA,JNBFPB, NSSCT,
     1 IX0,IY0,(KPOPO(I),I=1,NSSCT)
81    FORMAT(A1,A4,1X,A4,I5,1X,A5,A4,1X,I2,2I4,14I7)
      IY0 = IY
      IX0 = IX
      IXL = IX
      NSSCT = 1
      KPOPO(NSSCT) = IPNT(J)
      GO TO 92
93    CONTINUE
      IF(NSSCT .LT. NSSMX) GO TO 94
      JOUTY = JOUTY + 1
      WRITE(MU,81) MFR,MRTY,MRTY,JOUTY,JNBFPA,JNBFPB, NSSCT,
     1 IX0,IY0,(KPOPO(I),I=1,NSSCT)
      IY0 = IY
      IX0 = IX
      IXL = IX
      NSSCT = 1
      KPOPO(NSSCT) = IPNT(J)
      GO TO 92
94    CONTINUE
      IF(IY0 .EQ.IY) GO TO 95
      JOUTY = JOUTY + 1
      WRITE(MU,81) MFR,MRTY,MRTY,JOUTY,JNBFPA,JNBFPB, NSSCT,
     1 IX0,IY0,(KPOPO(I),I=1,NSSCT)
      IY0 = IY
      IX0 = IX
      IXL = IX
      NSSCT = 1
      KPOPO(NSSCT) = IPNT(J)
      GO TO 92
95    CONTINUE
      NSSCT = NSSCT + 1
      KPOPO(NSSCT) = IPNT(J)
      IXL = IX
92    CONTINUE
C
      IF(NINCT .LT.6) GO TO 62
60    CONTINUE
      JOUTCT = JOUTCT + 1
      WRITE(MT,51) MFR, MRCT,MRCT , JOUTCT, JNBFPA,JNBFPB,
     1 NINCT, (IARR(I),JARR(I),NPARR(I),I=1,6)
51    FORMAT(A1,A4,1X,A4,I5,1X,A5,A4,1X,I1,1X,6(I4,I4,I8,1X))
      DO 61 I = 1,6
      IARR(I) = 0
      JARR(I) = 0
      NPARR(I) = 0
```

Writ QBAA file

```
61      CONTINUE
        NINCT = 1
        IARR(NINCT) = IX
        JARR(NINCT) = IY
        NPARR(NINCT) = IPNT(J)
        GO TO 54
62      CONTINUE
        NINCT = NINCT + 1
        IARR(NINCT) = IX
        JARR(NINCT) = IY
        NPARR(NINCT) = IPNT(J)
        GO TO 54
C
53      CONTINUE
C
C       FINISH FOR THIS CITY
        IF(IEND .EQ. 1) GO TO 64
        ITMA = MRTH
        ITMB = MRTYH
        GO TO 65
64      CONTINUE
        ITMA = 4HSTOP
        ITMB = 4HSTOP
65      CONTINUE
        JOUTCT = JOUTCT + 1
        WRITE(MT,51) MFB, MRCT,ITMA , JOUTCT, JNBFPA,JNBFPB,
       1 NINCT, (IARR(I),JARR(I),NPARR(I),I=1,6)
        JOUTY = JOUTY + 1
        WRITE(MU,81) MFB,MRTY,ITMB,JOUTY,JNBFPA,JNBFPB, NSSCT,
       1 IXO,IYO,(KPOPO(I),I=1,NSSCT)
        IF(IEND .NE. 1) GO TO 20
C
C
200     CONTINUE
        WRITE(MO,201)
201     FORMAT(/////,20X,*END OF RUN*,///           )
        WRITE(MO,202)
202     FORMAT(1H1)
        ENDFILE MT
        ENDFILE MT
        ENDFILE MU
        ENDFILE MU
        ENDFILE MV
        ENDFILE MV
        STOP 6400
        END
 220 48441 0040      ABILENE
 170 39153 0080      AKRON
  56 13095 0120      ALBANY
 140 36001 0160      ALBANY-SCHENECTADY
 139 35001 0200      ALBUQUERQUE
 185 42077 0240      ALLENTOWN-BETHLEHE
 179 42013 0280      ALTOONA
 217 48375 0320      AMARILLO
  73 18095 0400      ANDERSON
 114 26161 0440      ANN ARBOR
 245 55087 0459      APPLETON
 147 37021 0480      ASHEVILLE
  57 13121 0520      ATLANTA
 136 34001 0560      ATLANTIC CITY
  59 13245 0600      AUGUSTA
```

End of UA.

Name, UA code association data

```
 63 17089 0620     AURORA-ELGIN
223 48453 0640     AUSTIN
 14 06029 0680     BAKERSFIELD
 97 24510 0720     BALTIMORE
 91 22033 0760     BATON ROUGE
109 26017 0800     BAY CITY
211 48245 0839     BEAUMONT
120 30111 0880     BILLINGS
121 28047 0920     BILOXI-GULFPORT
141 36007 0959     BINGHAMTON
  2 01073 1000     BIRMINGHAM
 64 17113 1040     BLOOMINGTON-NORMAL
 60 16001 1080     BOISE CITY
103 25017 1120     BOSTON
 23 08013 1130     BOULDER
 32 09001 1160     BRIDGEPORT
 34 09003 1170     BRISTOL
105 25023 1200     BROCKTON
201 48061 1239     BROWNSVILLE
200 48041 1260     BRYAN-COLLEGE STAT
142 36029 1280     BUFFALO
160 39151 1320     CANTON
 81 19113 1360     CEDAR RAPIDS
 61 17019 1400     CHAMPAIGN-URBANA
190 45019 1440     CHARLESTON
239 54039 1480     CHARLESTON
153 37119 1520     CHARLOTTE
195 47065 1560     CHATTANOOGA
 62 17031 1601     CHICAGO ILL-NORTHW
162 39061 1639     CINCINNATI
166 39035 1680     CLEVELAND
 30 08041 1720     COLORADO SPRINGS
123 29019 1740     COLUMBIA
192 45079 1760     COLUMBIA
 58 13215 1800     COLUMBUS
161 39049 1840     COLUMBUS
216 48355 1880     CORPUS CHRISTI
203 48113 1920     DALLAS
 33 09001 1930     DANBURY
 67 17161 1960     DAVENPORT-ROCK ISL
167 39113 2000     DAYTON
 65 17115 2040     DECATUR
 20 08031 2080     DENVER
 82 19153 2120     DES MOINES
117 26163 2160     DETROIT
 80 19061 2201     DUBUQUE
120 27137 2240     DULUTH-SUPERIOR
149 37063 2280     DURHAM
205 48141 2320     EL PASO
182 42049 2360     ERIE
174 41039 2400     EUGENE
 77 18163 2439     EVANSVILLE
 93 25005 2480     FALL RIVER
156 38017 2520     FARGO-MOORHEAD
148 37051 2560     FAYETTEVILLE
106 25027 2600     FITCHBURG-LEOMINST
110 26049 2640     FLINT
 45 12011 2680     FORT LAUDERDALE-HO
 11 05131 2720     FORT SMITH
 71 18003 2760     FORT WAYNE
219 48439 2800     FORT WORTH
```

```
 13 06019 3840      FRESNO
  1 01055 2880      GADSDEN
 44 12001 2900      GAINESVILLE
206 48167 2919      GALVESTON
113 26081 3000      GRAND RAPIDS
128 30013 3040      GREAT FALLS
240 55009 3080      GREEN BAY
151 37081 3119      GREENSBORO
191 45045 3160      GREENVILLE
158 39017 3189      HAMILTON
208 48061 3230      HARLINGEN-SAN BENI
181 42043 3240      HARRISBURG
 37 09003 3280      HARTFORD
152 37081 3300      HIGHPOINT
209 48201 3360      HOUSTON
237 54011 3400      HUNTINGTON-ASHLAND
  3 01089 3440      HUNTSVILLE
 74 18097 3480      INDIANAPOLIS
111 26075 3520      JACKSON
122 28049 3560      JACKSON
 47 12031 3600      JACKSONVILLE
180 42021 3680      JOHNSTOWN
 69 17197 3700      JOLIET
112 26077 3720      KALAMAZOO
126 29095 3760      KANSAS CITY
242 55059 3800      KENOSHA
196 47093 3840      KNOXVILLE
243 55063 3871      LACROSSE
 92 22055 3880      LAFAYETTE
 76 18157 3920      LAFAYETTE-WEST LAF
 90 22019 3960      LAKE CHARLES
184 42071 4000      LANCASTER
110 26065 4040      LANSING
223 48479 4080      LAREDO
132 32003 4120      LAS VEGAS
101 25009 4160      LAWRENCE-HAVERHILL
171 40031 4200      LAWTON
 95 23001 4240      LEWISTON-AUBURN
 87 21067 4280      LEXINGTON
157 39003 4320      LIMA
131 31109 4360      LINCOLN
 10 05119 4400      LITTLE ROCK-NORTH
164 39093 4440      LORAIN-ELYRIA
 15 06037 4480      L.A.-LONG BEACH
 88 21111 4520      LOUISVILLE
104 25017 4561      LOWELL
213 48303 4600      LUBBOCK
228 51680 4640      LYNCHBURG
 54 13021 4680      MACON
241 55025 4720      MADISON
134 33011 4760      MANCHESTER
168 39139 4800      MANSFIELD
214 48215 4880      MCALLEN-PHARR-EDIN
197 47157 4921      MEMPHIS
 39 09009 4960      MERIDEN
 46 12025 5000      MIAMI
215 48329 5040      MIDLAND
244 55079 5080      MILWAUKEE
118 27053 5120      MINNEAPOLIS-ST PAU
  4 01097 5160      MOBILE
 25 06099 5170      MODESTO
 94 22073 5200     MONROE
  5 01101 5240     MONTGOMERY
 72 18035 5280     MUNCIE
114 26121 5320     MUSKEGON-MUSKEGON
135 33011 5350     NASHUA
194 47037 5360     NASHVILLE-DAVIDSON
100 25005 5400     NEW BEDFORD
 39 09003 5440     NEW BRITAIN
 40 09009 5480     NEW HAVEN
 93 22071 5560     NEW ORLEANS
143 36047 5601     NEW YORK-N E NEW J
229 51700 5680     NEWPORT NEWS-HAMPT
230 51710 5720     NORFOLK-PORTSMOUTH
 34 09001 5760     NORWALK
204 48135 5800     ODESSA
227 49057 5840     OGDEN
172 40109 5880     OKLAHOMA CITY
130 31055 5920     OMAHA
 51 12095 5960     ORLANDO
247 55139 5970     OSHKOSH
 86 21059 5990     OWENSBORO
 26 06111 6000     OXNARD-VENTURA-100
 49 12033 6080     PENSACOLA
 64 17143 6120     PEORIA
231 51730 6140     PETERSBURG-COLONIA
187 42101 6160     PHILADELPHIA
  7 04013 6200     PHOENIX
  9 05069 6240     PINE BLUFF
177 42003 6280     PITTSBURGH
 99 25003 6320     PITTSFIELD
212 48245 6380     PORT ARTHUR
 96 23005 6400     PORTLAND
174 41051 6440     PORTLAND
189 44007 6480     PROVIDENCE-PAWTUCK
226 49049 6520     PROVO-OREM
 31 08101 6560     PUEBLO
246 55101 6600     RACINE
155 37183 6640     RALEIGH
179 42011 6680     READING
133 32031 6720     RENO
232 51760 6760     RICHMOND
233 51770 6800     ROANOKE
110 27109 6820     ROCHESTER
144 36055 6840     ROCHESTER
 76 17201 6880     ROCKFORD
 18 06067 6920     SACRAMENTO
115 26145 6960     SAGINAW
124 29021 7001     ST JOSEPH
127 29189 7040     ST LOUIS
 53 12103 7060     ST PETERSBURG
175 41047 7080     SALEM
 16 06053 7119     SALINAS
225 49035 7160     SALT LAKE CITY
221 48451 7200     SAN ANGELO
199 48029 7240     SAN ANTONIO
 10 06071 7279     SAN BERNARDINO-RIV
 20 06073 7320     SAN DIEGO
 12 06001 7360     SAN FRANCISCO-OAKL
 23 06085 7400     SAN JOSE
 22 06083 7480     SANTA BARBARA
 24 06097 7500     SANTA ROSA
 56 13051 7520   SAVANNAH
183 42069 7560   SCRANTON
 17 06053 7580   SEASIDE-MONTEREY
234 53033 7600   SEATTLE-EVERETT
209 48181 7640   SHERMAN-DENISON
 89 22017 7680   SHREEVEPORT
 27 06111 7700   SIMI VALLEY
 83 19193 7721   SIOUX CITY
193 46099 7760   SIOUX FALLS
 75 18141 7801   SOUTH BEND
236 53063 7840   SPOKANE
 68 17167 7880   SPRINGFIELD
125 29077 7920   SPRINGFIELD
159 39023 7960   SPRINGFIELD
102 25013 8000   SPRINGFIELD-CHICOP
 35 09001 8040   STAMFORD
163 39081 8080   STEUBENVILLE-WEIRT
 21 06077 8120   STOCKTON
146 36067 8160   SYRACUSE
235 53053 8200   TACOMA
 50 12073 8240   TALLAHASSEE
 49 12057 8279   TAMPA
 78 18167 8320   TERRE HAUTE
190 48037 8360   TEXARKANA
207 48167 8380   TEXAS CITY-LA MARQ
165 39095 8400   TOLEDO
 85 20177 8440   TOPEKA
139 34021 8481   TRENTON
  8 04019 8520   TUCSON
173 40143 8560   TULSA
  6 01125 8600   TUSCALOOSA
218 48423 8640   TYLER
145 36065 8680   UTICA-ROME
137 34011 8760   VINELAND-MILLVILLE
214 48309 8800   WACO
 43 11001 8840   WASHINGTON
 41 09009 8880   WATERBURY
 79 19013 8920   WATERLOO
 52 12099 8960   WEST PALM BEACH
239 54069 9000   WHEELING
 84 20173 9040   WICHITA
224 48485 9080   WICHITA FALLS
186 42079 9119   WILKES-BARRE
 42 10003 9160   WILMINGTON
154 37129 9200   WILMINGTON
150 37067 9230   WINSTON-SALEM
107 25027 9240   WORCESTER
188 42133 9290   YORK
166 39099 9320   YOUNGSTOWN-WARREN
```

COMPUTER PROGRAM DESCRIPTION

NAME: LLIST

SYNOPSIS: Special Purpose Listing

TYPE: SINGLE USE

USE: To list towns (urban nodes not urbanized areas) in order of decreasing standard deviation of population to assess nationwide distribution.

BACKGROUND:

DESCRIPTION: The input model file is read and urban records are stored in core. A sort on semi major axis is made and the output printed.

INPUT: Standard File XBAA-2

OUTPUT: Listing

STORAGE: IDA Card Deck S-10

DOCUMENTATION: Attached Listing
See Run S-49

LANGUAGE/SYSTEM: Run (FORTRAN)/6400 Scope

COMMENTS:

Preceding Page BLANK - NOT FILMED

```
      PROGRAM LLIST(TIN,OUTPUT)
C
C     TO LIST TOWNS IN ORDER OF DECREASING SIGB
C
      DIMENSION IFPA(2500),IFPB(2500),NAMA(2500),  NAMB(2500),
     1 NAMC(2500), IPCP(2500), FLAT(2500), FLON(2500), SIGB(2500),
     2  SIGL(2500), ALPH(2500), IND(2500)
      MT = 3LTIN
      MQ = 6LOUTPUT
      IDON = 0
      WRITE(MQ, 12)
12    FORMAT( 1H1,///////,20X,*LIST OF TOWN FROM XB  -1 (3675) ORDERED B
     1Y DECREAZING SEMI MAJOR AXIS*,//////////)
      I = 0
C
10    CONTINUE
      I = I + 1
       READ(MT,21)     IREC,IRECN,IFPA(I),IFPB(I), NAMA(I),NAMB(I),
     1 NAMC(I), IPOP(I),FLAT(I),FLON(I), SIGB(I),SIGL(I),ALPH(I)
21    FORMAT(1X,A4,1X,A4,6X,A5,A4,  1X,2A10,A4, I9,2F8.4,2F7.3,F7.2)
20    CONTINUE
      IF(IRECN .EQ. 4HSTOP) GO TO 50
      IF(IRECN .EQ.4HLB     ) GO TO 10
      READ (MT, 22) IRECN
22    FORMAT( 2X,A4)
      GO TO 20
C
50    CONTINUE
      INN = I
      CALL ORDER (SIGB,IND,INN    )
C
58    CONTINUE
      DO 60 J = 1,INN
      JK = INN - J + 1
      I = IND(JK)
      WRITE(MQ, 62)J,IFPA(I),IFPB(I),NAMA(I),NAMB(I),NAMC(I),
     1IPOP(I), FLAT(I),FLON(I), SIGB(I),SIGL(I),ALPH(I)
62    FORMAT( 1H ,I5,3X,A5,1X,A4,4X,2A10,A4,5X,I9, F8.4,1X,F8.4,
     1 6X,F7.3,2X,F7.3,3X,F7.2)
60    CONTINUE
      IF(IDON .EQ.1) GO TO 7
      WRITE(MQ,12)
      IDON = 1
      GO TO 58
70    CONTINUE
C
      STOP 6400
      END
```

Read number area records

Sort on semi major axis of population distribution.

Print ordered records.

Preceding Page BLANK - NOT FILMED

COMPUTER PROGRAM DESCRIPTION

NAME: WPGRID

SYNOPSIS: Map Weapons on an Urbanized Area

TYPE: SINGLE USE

USE: To produce maps to assist visualizing the location of weapons in attack on urbanized areas.

BACKGROUND: Modification of the program TOGRID to plot weapons.

DESCRIPTION: An attack file is input and stored. In each urbanized area the maximum population range is increased by 20 points. All weapons in the enlarged grid are associated with a grid point.

INPUT: Standard File GBBD
Attack Weapon File

OUTPUT: Plots of Weapons

STORAGE: IDA Card Deck 209

DOCUMENTATION: Attached Listing

LANGUAGE/SYSTEM: Run (FORTRAN)/6400 Scope

COMMENTS: Also outputs files of QBAA and QBAB format.

Preceding Page Blank - NOT FILMed

```
      PROGRAM WPGRID(TINA,TINB,TOUTA,TOUTC,OUTPUT,INPUT,TOUTB)
C
      DIMENSION IPNT(20000)
      DIMENSION IARR(6),JARR(6),NPARR(6),KPOPO(20)
      DIMENSION FLATI(4),FLONI(4),KPOPI(4)
      DIMENSION LNVAL(40),LNPRT(140) ,INUMB(10)
      DIMENSION NAMEA(250),NAMEB(250),IACAR(250)
      DIMENSION IUACW(2000),FLATW(2000),FLONW(2000),YLD(2000),
     1 ICLW(2000),IWAT(2000)
      DATA LNVAL     / 1H ,1H.,1H1,    1H2,1H3,1H4,1H5, 1H6,1H7,1H8,1H9,
     1 1H0,1HA,1HB,1HC,1HD,1HE,  1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,
     2 1HN,1HO, 1HP,1HQ,1HR,1HS,1HT,1HU,1HV,1HW,1HX,1HY,1HZ,1H+,1H*/
      DATA  INUMB / 1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H0  /
      MA = 4LTINA
      MB = 4LTINB
      MT = 5LTOUTA
      MU = 5LTOUTB
      MV = 5LTOUTC
      MQ = 6LOUTPUT
      MP = 5LINPUT
C     MA  IS HEADER ON GBBD, MB IS DATA(4 IN 1) ON GBBC
      MFB = 1H
      MFBO = 1H0
      MFBS = 1H1
      IEND = 0
      ISTRT = 1
      MRCT = 4HQWAA
      MRTH = 4HQWAB
      MRTY = 4HQWBA
      MRTYH = 4HQWBB
      MRTMP = 4HQWCA
      MRTMPH = 4HQWCB
      MSKA =           777777777777777777B
      MSKB =         33000000000000000000B
      MSKC =         77000000000000000000B
      MSKD =         55000000000000000000B
      IBO = 1
      IBO = 0
      ITOH = 1
      ITOH = 0
      IBUGCT  = 0
      IBUGA = 0
      IBUGCT = 9999
      IBUGA = 9999
      IBUGMX = 3
      IBUGMX = IBUGCT + 20
      IBUGMX = 99999
      ISKIP = 0
      JOUTY = 0
      JOUTMP = 0
      JOUTCT = 0
      NSSMX = 14
      SMIC = 69.17352282
      SPGX = 2.
      SPGY = 2.
      SPGX = 1.6666666667
      SPGXO = 24./SMIC
      SPGY = 24./SMIC
      SPGX = 1.
      SPGY = 1.
      SPGX = 0.4
```

Preceding Page BLANK - NOT FILMED

```
      SPGY = 0.4
      DISTX = 1./SPGX
      DISTY = 1./SPGY
      DISTHX = U.5*DISTX
      DISTHY = 0.5*DISTY
      WRITE(MQ,9 )  SPGY,SPGX
9     FORMAT(  1H1,//////,*                RUN OF GRID INPUT PROGRAM*,/////,
     1 *    IN THIS RUN THERE ARE *,F8.4, * GRID POINTS NORTH SOUTH AND
     2*,F8.4,* GRID POINTS EAST WEST IN EACH MILE*,///)
      INN = 0
11    CONTINUE
      INN = INN + 1
      READ(MP,12) IACAR(INN),NAMEA(INN),NAMEB(INN)
12    FORMAT(1X,A4,5X,A10,A8)
      IF(IACAR(INN) .NE. 4H9320) GO TO 11
      J = 0
401   CONTINUE
      J = J + 1
      READ(MB,402) ITTM    ,FLOD,FLOM,FLOS,FLAD,FLAM,FLAS,YLD(J),ICLW(J)
     1 , IWAT(J)
402   FORMAT(1X,A4,40X,6F4.0, F4.0,3X,I4,1X,A2)
      FLONW(J)= FLOD + FLOM/60. + FLOS/3600.
      FLATW(J) = FLAD + FLAM/60. + FLAS/3600.
      ITM = ITTM .AND. MSKC
      IF(ITM .NE. MSKD) GO TO 404
      ITTM = MSKB .OR. (ITTM .AND. MSKA)
404   CONTINUE
      IUACW(J) = ITTM
      IF(ICLW(J) .NE. 5602) GO TO 401
      INW = J
C
20    CONTINUE
      IBUGCT = IBUGCT + 1
      IF(IBUGCT .GT. IBUGMX) IEND = 1
      DO 14 I = 1,140
      LNPRT(I) = 1H
14    CONTINUE
      READ(MA,21) MFBNX,IRGD,IRGDN,ICNT,KNBFPA,KNBFPB, IPOPT
     1 ,FLATCG,FLONCG,FLATMN,FLATMX,FLONMN,FLONMX   ,NUREC,KCENT
     2,SIGB,SIGL,ALPH
21    FORMAT(A1,A4,1X,A4,   A5,1X,A5,A4,1X,   I9,2F8.4,1X,2F8.4,1X,
     1 2F8.4 I5,1X,A5, 1X,2F7.4,F7.2)
      IF(EOF,MA) 22,23
22    CONTINUE
      GO TO 200
23    CONTINUE
      IF(IRGDN .EQ.4HSTOP) IEND = 1
      JNBFPA = KCENT
      JNBFPB = KNBFPB
      DO 25 I = 1,INN
      IF(KNBFPB .EQ. IACAR(I)) GO TO 26
25    CONTINUE
      NAMEAP = 10HUNLISTED
      NAMEBP = 8H
      GO TO 27
26    CONTINUE
      NAMEAP = NAMEA(I)
      NAMEBP = NAMEB(I)
27    CONTINUE
C     SPECIAL MODS FOR 2 1/2 MINUTE GRID*********************** START
      IF(ITOH .NE.1 ) GO TO  475
```

Input O.A. names

Read Weapons

Read new urbanized area

3

```
      IFLA = FLATCG*24.
      IFLO = FLONCG*24.
      IFLA = IFLA + 1
      IFLO = IFLO + 1
      FLATCG = IFLA
      FLONCG = IFLO
      FLATCG = FLATCG/24.
      FLONCG = FLONCG/24.
      SPGX = SPGXO/(COS(FLATCG*3.14159265/180.))
      DISTX = 1./SPGX
      DISTHX = .5*DISTX
475   CONTINUE
C     SPECIAL MODS FOR 2 1/2 MINUTE GRID********************** END
      DY = SMIC
      DX = SMIC*COS(FLATCG*3.14159265/180.)
      NGY = (FLATCG-FLATMN)*DY/DISTY+ DISTHY
      NGX = (FLONMX - FLONCG )*DX/DISTX+ DISTHX
      NGX = NGX + 20
      NGY = NGY + 20
      XLX = NGX*DISTX
      XLY = NGY*DISTY
      ITM = (FLATMX - FLATCG)*DY/DISTY+ DISTHY
      MAXY = NGY + ITM + 1
      ITM =(FLONCG - FLONMN)*DX/DISTX+ DISTHX
      MAXX = NGX + ITM + 1
      MAXX = MAXX + 20
      MAXY = MAXY + 20
      IF(MAXX .GT. 131) MAXX = 131
      IF(IY .GT. 150) IY = 150
      ISKIP = 0
      ITM = MAXX*MAXY
      IF(ITM .LT.20000) GO TO 106
      WRITE(MQ,107) KNBFPA,KNBFPA,MAXX,MAXY
107   FORMAT(      1H0,*ARRAY TOO SMALL SO SKIP AT*, A5,A4,2X,2I8    )
      ISKIP = 1
106   CONTINUE
C*******BUGGGGGGG
      IF(IBUGCT .GT. 10) GO TO 318
      WRITE(MQ,310)  DX,DY,XLX,XLY, NGX,NGY,MAXX,MAXY
310   FORMAT(1H ,* AT 310*,  4F12.4,4I8)
318   CONTINUE
      NWPC = 0
      XMTSM = 0.
      EMTSM = 0.
      DO 24 I = 1,MAXX
      DO 24 J = 1,MAXY
      K = I + MAXX*(J-1)
      IPNT(K) = 0
24    CONTINUE
      NNZ = 0
C
30    CONTINUE
      DO 39 I = 1,INW
      IF(IUACW(I) .NE. KNBFPB) GO TO 39
      IBUGA = IBUGA + 1
      IX = ((FLONCG- FLONW(I)) *DX + XLX)/DISTX + DISTHX
      IX = IX + 1
      IY = ((FLATW(I) - FLATCG)*DY + XLY)/DISTY + DISTHY
      IY = IY + 1
      ITROB = 0
      IF(IX .LE. MAXX .AND. IY .LE. MAXY) GO TO 41
```

Set grid

- enlarge grid

Search Weapons and use those with same U.A. code - Put weapon in grid

Get weapons inside U.A. grid.

```
      WRITE(MQ,40) KNBFPA,KNBFPB, IX,IY,YLD(I),ICLW(I),IWAT(I),FLATW(I),
     1 FLATW(I)
40    FORMAT( 1H ,* TOO LARGE AT * ,A5,1X,A4, * AT GRID X/Y = *,2I4,
     1 *  WPN YLD/CL NO./TYPE = *, F6.1, I5,1X,A2, * AT LAT/LON = *,
     2 2F9.4)
      IF(IX .GT. MAXX) IX = MAXX
      IF(IY.GT.MAXY) IY = MAXY
      ITROB = 1
41    CONTINUE
      IF(IX .GE.1.AND. IY.GE.1) GO TO 101
      WRITE(MQ,102)KNBFPA,KNBFPB, IX,IY,YLD(I),ICLW(I),IWAT(I),FLATW(I),
     1 FLATW(I)
102   FORMAT( 1H ,* TOO SMALL AT * ,A5,1X,A4, * AT GRID X/Y = *,2I4,
     1 *  WPN YLD/CL NO./TYPE = *, F6.1, I5,1X,A2, * AT LAT/LON = *,
     2 2F9.4)
      IF(IX .LT. 1) IX = 1
      IF(IY .LT. 1) IY = 1
      ITROB = 1
101   CONTINUE
C*****BUGGGGGGGG
      IF(IBUGCT .GT.1) GO TO 44
      WRITE(MQ,315)KCNT,MNBFPA,MNBFPB, IX,IY,MAXX,MAXY
315   FORMAT(1H0 *---   AT 315 NOW E, AT*,I5,  2X,A5,A4,4I9)
C*******BUGGGGGGG
      WRITE(MQ, 312) FLONCG,FLONI(I),FLATI(I),FLATCG, XLX,XLY
312   FORMAT( 1H  ,* AT 312  *, 6F12.4)
44    CONTINUE
      NWPC = NWPC + 1
      XMTSM = XMTSM + YLD(I)
      EMT = YLD(I)**0.6666666667
      EMTSM = EMTSM + EMT
      J = IX +(IY - 1)*MAXX
      IF(IPNT(J) .EQ.0) GO TO 42
      IPNT(J) = IPNT(J) +  EMT
      GO TO 43
42    CONTINUE
      NNZ = NNZ + 1
      IPNT(J) = EMT
43    CONTINUE
      IF(ITROB .EQ.1) IPNT(J) = 99
39    CONTINUE
C
C
C     AT THI S POINT ARRAY S FILLED NOW START OUTPUT
C
C     FIRST HEADERS FOR THE CITY
C
45    CONTINUE
C*****BUGGGGGGGG
      IF(IBUGCT .GT. 1) GO TO 343
      DO 340 J = 1,MAXY
      DO 340 I = 1,MAXX
      K = I + (J-1)*MAXX
      WRITE(MQ,342) I,J,K, IPNT(K)
342   FORMAT(1H  ,* ARRAAY PT =*,3I5,5X,I10)
340   CONTINUE
343   CONTINUE
      IF(ISKIP.EQ.1) GO TO 20
      JOUTCT = JOUTCT + 1
      WRITE(MT,46) MFBO, MRTH,MRCT , JOUTCT,JNBFPA,JNBFPB, IPOPT,
     1 FLATCG,FLONCG, NGY,NGX,XLY,XLX,MAXY,MAXX
```

4

Bring [illegible] out of range and put them to edge of grid

Find equivalent MT association

Output Map Header.

5

```
46     FORMAT(A1 ,A4,1X,A4,I5,1X,A5,A4,1X, I9,2F8.4,1X,2I4,1X,2F8.4,
      1 1X,2I4)
       IF(IBO .NE. 1 ) GO TO 431
       JOUTY = JOUTY + 1
       WRITE(MU,47) MFBO,MRTYH,MRTY, JOUTY, JNBFPA,JNBFPB, IPOPT ,
      1 FLATCG,FLONCG, NGY,NGX,XLY,XLX,MAXY,MAXX
47     FORMAT(A1 ,A4,1X,A4,I5,1X,A5,A4,1X, I9,2F8.4,1X,2I4,1X,2F7.4,
      1 1X,2I4)
431    CONTINUE
       JOUTMP = JOUTMP + 1
       WRITE( MV,48) MFBS,MRTMPH,MRTMP, JOUTMP, JNBFPA,JNBFPB,
      1 NAMEAP,NAMEBP,
      2 JNBFPB,JNBFPA, IPOPT, FLATCG,FLONCG, NGY,NGX, XLY,XLX,
      3 MAXY,MAXX
48     FORMAT( A1, A4,1X,A4,I5,1X,A5,A4,1X,///, 20X, *POPULATION GRID MAP
      1 FOR THE CITY OF *,A10,A8    ,//,          *  MAP FOR URBANIZED AREA
      2 NO. *,A4,* WITH STATE COUNTY CODE *,A5,* , AND POPULATION = *,I9,
      3 /,* LATITUDE/LONGITUDE OF C.G. OF POPULATION = *, F8.4,2X,F8.4,
      4  /,* GRID STEPS SOUTH /WEST OF C.G. = *,2I6, * AT DISTANCES OF *,
      5 2F9.4      ,/, * N/S GRID STEPS = *, I6,*  AND E/W GRID STEPS =*,
      6 I6)
       WRITE(MV,442)NWPC, NNZ,XMTSM,EMTSM
442    FORMAT(1H0, * FOR THIS CITY NUMBER OF WEAPONS = *,I4, * AT *,I4,
      1  * GRID POINTS, TOTAL MEGATONS =*, F9.2,* AND EQUIVALENT MEGATONS
      2 =*,F9.2 ,//)
       DO 82 I = 1,MAXX
       II = I/10
       IA = I - II*10
       IF(IA .EQ. 0) GO TO 83
       LNPRT(I) = 1H+
       GO TO 82
83     CONTINUE
       IF(II .GT. 10) GO TO 84
       LNPRT(I) = INUMB(II)
       GO TO 82
84     CONTINUE
       II = II - 10
       GO TO 83
82     CONTINUE
       LNPRT(NGX +  1) = 1H*
       WRITE(MV,85)(LNPRT(I),I = 1,MAXX   )
85     FORMAT( 1H ,3X, 131A1)
C
C
C      NOW START AT TOP LINE AND WORK DOWN
C
C
C
       NINCT = 0
       NSSCT = 0
       IY = MAXY + 1
55     CONTINUE
C      NEW LINE
       IX = 0
54     CONTINUE
       IX = IX + 1
       IF(IX .LE.MAXX) GO TO 52
       ITM = MAXY + 1
       IF(IY .EQ. ITM) GO TO 72
       JOUTMP = JOUTMP + 1
       ITM = NGY +  1
       IF(IY .NE. ITM) GO TO 406
```

Output Map

6

```
       WRITE(MV,407)(LNPRT(I),I = 1,MAXX    )
407    FORMAT(1H  ,2X,1H*, 131A1)
       GO TO 87
406    CONTINUE
       ITTM = MAXY - IY
       JPR = ITTM /10
       ITM = ITTM - JPR*10
       IF(ITM .EQ. 0) GO TO 86
       WRITE(MV,88) (LNPRT(I),I = 1,MAXX    )
88     FORMAT(1H  ,2X,1H+, 131A1)
       GO TO 87
86     CONTINUE
       WRITE(MV,73) JPR,(LNPRT(I), I = 1,MAXX)
73     FORMAT(1H ,I3,      131A1)
87     CONTINUE
72     CONTINUE
       IY = IY - 1
       IF(IY .LT. 1) GO TO 53
       GO TO 55
52     CONTINUE
       J = (IY - 1)*MAXX + IX
       IF(IPNT(J) .NE.0) GO TO 74
       JTTM = MAXX - 19
       JTM = MAXY - 19
       IF(IY .NE. 20 .AND. IY .NE. JTM) GO TO 411
       IF(IX .LT. 20 .OR. IX .GT. JTTM ) GO TO 411
       LNPRT(IX) = 1H.
       GO TO 412
411    CONTINUE
       IF(IX .NE. 20 .AND. IX. NE. JTTM) GO TO 413
       IF(IY .LT. 20 .OR. IY .GT. JTM) GO TO 413
       LNPRT(IX) = 1H.
       GO TO  412
413    CONTINUE
       LNPRT(IX) = 1H
412    CONTINUE
       GO TO 408
74     CONTINUE
       K = IPNT(J)*2
C*****BUGGGGGGGG
       IF(K .GE. 0) GO TO 632
       WRITE(MG,633) K   ,IX,IY,J,IPNT(J)
633    FORMAT(1H   ,* AT LOI WITH NEG K*,  5I10)
       K = 0
632    CONTINUE
       K = K + 2
       IF(K .LE. 38) GO TO 71
       K = 40
71     CONTINUE
       LNPRT(IX) = LNVAL(K)
408    CONTINUE
C
       IF(IPNT(J) .EQ.0) GO TO 54
       IF(IBO .NE. 1 ) GO TO 430
       IF(NSSCT .NE. 0) GO TO 91
       NSSCT = 1
       KPOPO(NSSCT) = IPNT(J)
       IYO = IY
       IXO = IX
       IXL = IX
       GO TO 92
```

```
91    CONTINUE
      ITM = IXL + 1
      IF(ITM .EQ.IX) GO TO 93
      JOUTY = JOUTY + 1
      WRITE(MU,81) MF9,MRTY,MRTY,JOUTY,JNBFPA,JNBFPR, NSSCT,
     1 IXO,IYO,(KPOPO(I),I=1,NSSCT)
81    FORMAT(A1,A4,1X,A4,I5,1X,A5,A4,1X,I2,2I4,14I7)
      IYO = IY
      IXO = IX
      IXL = IX
      NSSCT = 1
      KPOPO(NSSCT) = IPNT(J)
      GO TO 92
93    CONTINUE
      IF(NSSCT .LT. NSSMX) GO TO 94
      JOUTY = JOUTY + 1
      WRITE(MU,81) MF9,MRTY,MRTY,JOUTY,JNBFPA,JNBFPR, NSSCT,
     1 IXO,IYO,(KPOPO(I),I=1,NSSCT)
      IYO = IY
      IXO = IX
      IXL = IX
      NSSCT = 1
      KPOPO(NSSCT) = IPNT(J)
      GO TO 92
94    CONTINUE
      IF(IYO .EQ.IY) GO TO 95
      JOUTY = JOUTY + 1
      WRITE(MU,81) MF9,MRTY,MRTY,JOUTY,JNBFPA,JNBFPR, NSSCT,
     1 IXO,IYO,(KPOPO(I),I=1,NSSCT)
      IYO = IY
      IXO = IX
      IXL = IX
      NSSCT = 1
      KPOPO(NSSCT) = IPNT(J)
      GO TO 92
95    CONTINUE
      NSSCT = NSSCT + 1
      KPOPO(NSSCT) = IPNT(J)
      IXL = IX
92    CONTINUE
430   CONTINUE
C
      IF(NINCT .LT.6) GO TO 62
60    CONTINUE
      JOUTCT = JOUTCT + 1
      WRITE(MT,51) MF9, MRCT,MRCT , JOUTCT, JNBFPA,JNBFPB,
     1 NINCT, (IARR(I),JARR(I),NPARR(I),I=1,6)
51    FORMAT(A1,A4,1X,A4,I5,1X,A5,A4,1X,I1,1X,6(I4,I4,I8,1X))
      DO 61 I = 1,6
      IARR(I) = 0
      JARR(I) = 0
      NPARR(I) = 0
61    CONTINUE
      NINCT = 1
      IARR(NINCT) = IX
      JARR(NINCT) = IY
      NPARR(NINCT) = IPNT(J)
      GO TO 54
62    CONTINUE
      NINCT = NINCT + 1
      IARR(NINCT) = IX
```

Output 2BAR
format file

Output 2BRF
format file

```
      JARR(NINCT) = IY
      NPARR(NINCT) = IPNT(J)
      GO TO 54
C
53    CONTINUE
C
C     FINISH FOR THIS CITY
      IF(IEND .EQ.1) GO TO 64
      ITMA = MRTH
      ITMB = MRTYH
      GO TO 65
64    CONTINUE
      ITMA = 4HSTOP
      ITMB = 4HSTOP
65    CONTINUE
      JOUTCT = JOUTCT + 1
      WRITE(MT,51) MFB, MRCT,ITMA , JOUTCT, JNBFPA,JNBFPB,
     1 NINCT, (IARR(I),JARR(I),NPARR(I),I=1,6)
      IF(IBO .NE. 1 ) GO TO 432
      JOUTY = JOUTY + 1
      WRITE(MU,81) MFB,MRTY,ITMB,JOUTY,JNBFRA,JNBFPB, NSSCT,
     1 IXO,IYO,(KPOPO(I),I=1,NSSCT)
432   CONTINUE
      IF(IEND .NE. 1) GO TO 20
```

End of U.A

```
C
C
200   CONTINUE
      WRITE(MQ,201)
201   FORMAT(/////,20X,*END OF RUN*,///         )
      WRITE(MQ,202)
202   FORMAT(1H1)
      ENDFILE MT
      ENDFILE MT
      IF(IBO .NE. 1 ) GO TO 433
      ENDFILE MU
      ENDFILE MU
433   CONTINUE
      ENDFILE MV
      ENDFILE MV
      STOP 6400
      END
```

U.A code, U.A name
data file.

```
 220 48441 0040       ABILENE
 170 39153 0080       AKRON
  56 13095 0120       ALBANY
 140 36001 0160       ALBANY-SCHENECTADY
 139 35001 0200       ALBUQUERQUE
 185 42077 0240       ALLENTOWN-BETHLEHE
 179 42013 0280       ALTOONA
 217 48375 0320       AMARILLO
  73 18095 0400       ANDERSON
 116 26161 0440       ANN ARBOR
 245 55087 0459       APPLETON
 147 37021 0480       ASHEVILLE
  57 13121 0520       ATLANTA
 136 34001 0560       ATLANTIC CITY
  59 13245 0600       AUGUSTA
  63 17089 0620       AURORA-ELGIN
 222 48453 0640       AUSTIN
  14 06029 0680       BAKERSFIELD
  97 24510 0720       BALTIMORE
  91 22033 0760       BATON ROUGE
```

9

```
108 26017 0800   BAY CITY
211 48245 0839   BEAUMONT
129 30111 0880   BILLINGS
121 28047 0920   BILOXI-GULFPORT
141 36007 0959   BINGHAMTON
  2 01073 1000   BIRMINGHAM
 64 17113 1040   BLOOMINGTON-NORMAL
 60 16001 1080   BOISE CITY
103 25017 1120   BOSTON
 28 08013 1130   BOULDER
 32 09001 1160   BRIDGEPORT
 36 09003 1170   BRISTOL
105 25023 1200   BROCKTON
201 48061 1239   BROWNSVILLE
200 48041 1260   BRYAN-COLLEGE STAT
142 36029 1280   BUFFALO
169 39151 1320   CANTON
 81 19113 1360   CEDAR RAPIDS
 61 17019 1400   CHAMPAIGN-URBANA
190 45019 1440   CHARLESTON
238 54039 1480   CHARLESTON
153 37119 1520   CHARLOTTE
195 47065 1560   CHATTANOOGA
 62 17031 1601   CHICAGO ILL-NORTHW
162 39061 1639   CINCINNATI
160 39035 1680   CLEVELAND
 30 08041 1720   COLORADO SPRINGS
123 29019 1740   COLUMBIA
192 45079 1760   COLUMBIA
 58 13215 1800   COLUMBUS
161 39049 1840   COLUMBUS
216 48355 1880   CORPUS CHRISTI
203 48113 1920   DALLAS
 33 09001 1930   DANBURY
 67 17161 1960   DAVENPORT-ROCK ISL
167 39113 2000   DAYTON
 65 17115 2040   DECATUR
 29 08031 2080   DENVER
 82 19153 2120   DES MOINES
117 26163 2160   DETROIT
 82 19153 2120   DES MOINES
117 26163 2160   DETROIT
 80 19061 2201   DUBUQUE
120 27137 2240   DULUTH-SUPERIOR
149 37063 2280   DURHAM
205 48141 2320   EL PASO
182 42049 2360   ERIE
174 41039 2400   EUGENE
 77 18163 2439   EVANSVILLE
 99 25005 2480   FALL RIVER
156 38017 2520   FARGO-MOORHEAD
148 37051 2560   FAYETTEVILLE
106 25027 2600   FITCHBURG-LEOMINST
109 26049 2640   FLINT
 45 12011 2680   FORT LAUDERDALE-HO
 11 05131 2720   FORT SMITH
 71 18003 2760   FORT WAYNE
219 48439 2800   FORT WORTH
 13 06019 2840   FRESNO
  1 01055 2880   GADSDEN
 44 12001 2900   GAINESVILLE
```

I-287

10

```
206 48167 2919          GALVESTON
113 26081 3000          GRAND RAPIDS
128 30013 3040          GREAT FALLS
240 55009 3080          GREEN BAY
151 37081 3119          GREENSBORO
191 45045 3160          GREENVILLE
158 39017 3199          HAMILTON
202 48061 3230          HARLINGEN-SAN BENI
181 42043 3240          HARRISBURG
 37 09003 3280          HARTFORD
152 37081 3300          HIGHPOINT
209 48201 3360          HOUSTON
237 54011 3400          HUNTINGTON-ASHLAND
  3 01089 3440          HUNTSVILLE
 74 18097 3480          INDIANAPOLIS
111 26075 3520          JACKSON
122 28049 3560          JACKSON
 47 12031 3600          JACKSONVILLE
180 42021 3680          JOHNSTOWN
 69 17197 3700          JOLIET
112 26077 3720          KALAMAZOO
126 29095 3760          KANSAS CITY
242 55059 3800          KENOSHA
196 47093 3840          KNOXVILLE
243 55063 3871          LACROSSE
 92 22055 3880          LAFAYETTE
 76 18157 3920          LAFAYETTE-WEST LAF
 90 22019 3960          LAKE CHARLES
184 42071 4000          LANCASTER
110 26065 4040          LANSING
223 48479 4080          LAREDO
132 32003 4120          LAS VEGAS
101 25009 4160          LAWRENCE-HAVERHILL
171 40031 4200          LAWTON
 95 23001 4240          LEWISTON-AUBURN
 87 21067 4280          LEXINGTON
157 39003 4320          LIMA
131 31109 4360          LINCOLN
 10 05119 4400          LITTLE ROCK-NORTH
164 39093 4440          LORAIN-ELYRIA
 15 06037 4480          L.A.-LONG BEACH
 88 21111 4520          LOUISVILLE
104 25017 4561          LOWELL
213 48303 4600          LUBBOCK
228 51680 4640          LYNCHBURG
 54 13021 4680          MACON
241 55025 4720          MADISON
134 33011 4760          MANCHESTER
168 39139 4800          MANSFIELD
210 48215 4880          MCALLEN-PHARR-EDIN
197 47157 4921          MEMPHIS
 39 09009 4960          MERIDEN
 46 12025 5000          MIAMI
215 48329 5040          MIDLAND
244 55079 5080          MILWAUKEE
118 27053 5120          MINNEAPOLIS-ST PAU
  4 01097 5160          MOBILE
 25 06099 5170          MODESTO
 94 22073 5200          MONROE
  5 01101 5240          MONTGOMERY
 72 18035 5280          MUNCIE
```

//

```
114 26121 5320     MUSKEGON-MUSKEGON
135 33011 5350     NASHUA
194 47037 5360     NASHVILLE-DAVIDSON
100 25005 5400     NEW BEDFORD
 38 09003 5440     NEW BRITAIN
 40 09009 5480     NEW HAVEN
 93 22071 5560     NEW ORLEANS
143 36047 5601     NEW YORK-N E NEW J
229 51700 5680     NEWPORT NEWS-HAMPT
230 51710 5720     NORFOLK-PORTSMOUTH
 34 09001 5760     NORWALK
204 48135 5800     ODESSA
227 49057 5840     OGDEN
172 40109 5880     OKLAHOMA CITY
130 31055 5920     OMAHA
 51 12095 5960     ORLANDO
247 55139 5970     OSHKOSH
 86 21059 5990     OWENSBORO
 26 06111 6000     OXNARD-VENTURA-100
 48 12033 6080     PENSACOLA
 66 17143 6120     PEORIA
231 51730 6140     PETERSBURG-COLONIA
187 42101 6160     PHILADELPHIA
  7 04013 6200     PHOENIX
  9 05069 6240     PINE BLUFF
177 42003 6280     PITTSBURGH
 98 25003 6320     PITTSFIELD
212 48245 6380     PORT ARTHUR
 96 23005 6400     PORTLAND
176 41051 6440     PORTLAND
189 44007 6480     PROVIDENCE-PAWTUCK
226 49049 6520     PROVO-OREM
 31 08101 6560     PUEBLO
246 55101 6600     RACINE
155 37183 6640     RALEIGH
178 42011 6680     READING
133 32031 6720     RENO
232 51760 6760     RICHMOND
233 51770 6800     ROANOKE
119 27109 6820     ROCHESTER
144 36055 6840     ROCHESTER
 70 17201 6880     ROCKFORD
 18 06067 6920     SACRAMENTO
115 26145 6960     SAGINAW
124 29021 7001     ST JOSEPH
127 29189 7040     ST LOUIS
 53 12103 7060     ST PETERSBURG
175 41047 7080     SALEM
 16 06053 7119     SALINAS
225 49035 7160     SALT LAKE CITY
221 48451 7200     SAN ANGELO
198 48029 7240     SAN ANTONIO
 19 06071 7279     SAN BERNARDINO-RIV
 20 06073 7320     SAN DIEGO
 12 06001 7360     SAN FRANCISCO-OAKL
 23 06085 7400     SAN JOSE
 22 06083 7480     SANTA BARBARA
 24 06097 7500     SANTA ROSA
 55 13051 7520     SAVANNAH
183 42069 7560     SCRANTON
 17 06053 7580     SEASIDE-MONTEREY
```

```
114 26121 5320    MUSKEGON-MUSKEGON
135 33011 5350    NASHUA
194 47037 5360    NASHVILLE-DAVIDSON
100 25005 5400    NEW BEDFORD
 38 09003 5440    NEW BRITAIN
 40 09009 5480    NEW HAVEN
 93 22071 5560    NEW ORLEANS
143 36047 5601    NEW YORK-N E NEW J
229 51700 5680    NEWPORT NEWS-HAMPT
230 51710 5720    NORFOLK-PORTSMOUTH
 34 09001 5760    NORWALK
204 48135 5800    ODESSA
227 49057 5840    OGDEN
172 40109 5880    OKLAHOMA CITY
130 31055 5920    OMAHA
 51 12095 5960    ORLANDO
247 55139 5970    OSHKOSH
 86 21059 5990    OWENSBORO
 26 06111 6000    OXNARD-VENTURA-100
 48 12033 6080    PENSACOLA
 66 17143 6120    PEORIA
231 51730 6140    PETERSBURG-COLONIA
187 42101 6160    PHILADELPHIA
  7 04013 6200    PHOENIX
  9 05069 6240    PINE BLUFF
177 42003 6280    PITTSBURGH
 98 25003 6320    PITTSFIELD
212 48245 6380    PORT ARTHUR
 96 23005 6400    PORTLAND
176 41051 6440    PORTLAND
189 44007 6480    PROVIDENCE-PAWTUCK
226 49049 6520    PROVO-OREM
 31 08101 6560    PUEBLO
246 55101 6600    RACINE
155 37183 6640    RALEIGH
178 42011 6680    READING
133 32031 6720    RENO
232 51760 6760    RICHMOND
233 51770 6800    ROANOKE
119 27109 6820    ROCHESTER
144 36055 6840    ROCHESTER
 70 17201 6880    ROCKFORD
 18 06067 6920    SACRAMENTO
115 26145 6960    SAGINAW
124 29021 7001    ST JOSEPH
127 29189 7040    ST LOUIS
 53 12103 7060    ST PETERSBURG
175 41047 7080    SALEM
 16 06053 7119    SALINAS
225 49035 7160    SALT LAKE CITY
221 48451 7200    SAN ANGELO
198 48029 7240    SAN ANTONIO
 19 06071 7279    SAN BERNARDINO-RIV
 20 06073 7320    SAN DIEGO
 12 06001 7360    SAN FRANCISCO-OAKL
 23 06085 7400    SAN JOSE
 22 06083 7480    SANTA BARBARA
 24 06097 7500    SANTA ROSA
 55 13051 7520    SAVANNAH
183 42069 7560    SCRANTON
 17 06053 7580    SEASIDE-MONTEREY
```

Documentation of Current IDA Computer Material Developed for DCPA, by Leo A. Schmidt, Unclassified, Institute for Defense Analyses, January 1977, Vol. I--283 pages, (Contract DCPA01-76-C-0213, Work Unit 4126G)

Abstract

This paper is a documentation of computer materials developed by the Institute for Defense Analyses (IDA) for use by the Defense Civil Preparedness Agency (DCPA). All IDA physical data processing materials (IBM cards, magnetic tape, computer printouts) have been surveyed and catalogued. All computer programs are written in FORTRAN (a general knowledge of this language is assumed in the detailed descriptions contained herein). Computer programs considered useful by IDA have been included and documented. A group of general purpose subprograms are described, along with their interfaces with the using programs. Data file formats also have been developed, along with programs for managing these files. Such programs and resulting files are described in detail.

Documentation of Current IDA Computer Material Developed for DCPA, by Leo A. Schmidt, Unclassified, Institute for Defense Analyses, January 1977, Vol. I--283 pages, (Contract DCPA01-76-C-0213, Work Unit 4126G)

Abstract

This paper is a documentation of computer materials developed by the Institute for Defense Analyses (IDA) for use by the Defense Civil Preparedness Agency (DCPA). All IDA physical data processing materials (IBM cards, magnetic tape, computer printouts) have been surveyed and catalogued. All computer programs are written in FORTRAN (a general knowledge of this language is assumed in the detailed descriptions contained herein). Computer programs considered useful by IDA have been included and documented. A group of general purpose subprograms are described, along with their interfaces with the using programs. Data file formats also have been developed, along with programs for managing these files. Such programs and resulting files are described in detail.

Documentation of Current IDA Computer Material Developed for DCPA, by Leo A. Schmidt, Unclassified, Institute for Defense Analyses, January 1977, Vol. I--283 pages, (Contract DCPA01-76-C-0213, Work Unit 4126G)

Abstract

This paper is a documentation of computer materials developed by the Institute for Defense Analyses (IDA) for use by the Defense Civil Preparedness Agency (DCPA). All IDA physical data processing materials (IBM cards, magnetic tape, computer printouts) have been surveyed and catalogued. All computer programs are written in FORTRAN (a general knowledge of this language is assumed in the detailed descriptions contained herein). Computer programs considered useful by IDA have been included and documented. A group of general purpose subprograms are described, along with their interfaces with the using programs. Data file formats also have been developed, along with programs for managing these files. Such programs and resulting files are described in detail.

Documentation of Current IDA Computer Material Developed for DCPA, by Leo A. Schmidt, Unclassified, Institute for Defense Analyses, January 1977, Vol. I--283 pages, (Contract DCPA01-76-C-0213, Work Unit 4126G)

Abstract

This paper is a documentation of computer materials developed by the Institute for Defense Analyses (IDA) for use by the Defense Civil Preparedness Agency (DCPA). All IDA physical data processing materials (IBM cards, magnetic tape, computer printouts) have been surveyed and catalogued. All computer programs are written in FORTRAN (a general knowledge of this language is assumed in the detailed descriptions contained herein). Computer programs considered useful by IDA have been included and documented. A group of general purpose subprograms are described, along with their interfaces with the using programs. Data file formats also have been developed, along with programs for managing these files. Such programs and resulting files are described in detail.